Web services guide v4.1

Table of contents

Overview

Watermark Faculty Success supports interoperability and automation via web services, including visibility into your data source's configuration, data import and export, user setup and management, and security role assignment. Each service resource supports a similar grammar and style, so the knowledge gained implementing a web service client for one resource transfers easily to others

https://support.watermarkinsights.com/hc/en-us/articles/4409232214683-REST-Style-Services

Getting started

1. Ask your Watermark Faculty Success administrator to request a Service Account with appropriate privileges via a work request.
2. Review the web services documentation. View the Schema List Resource to find your Schema Key(s), which is used by many other service resources.
3. Consider installing the RESTClient such as Postman (a simple interface for testing readable and writeable REST web services), available at
4. Download Postman | Get Started for Free
5. Find appropriate HTTPS Connection sample code for your development platform.
6. If you will be creating a web application based on web service data, choose an appropriate technology to transform XML data into HTML for your development platform. XSLT is one common option available on many platforms; using XPath to navigate a Document Object Model is another.
7. Read the documentation for the resources you plan on consuming. The sample use cases in this documentation can provide a starting point for a variety of tasks.
8. If you use resources that modify data, please submit your data to the corresponding Validate resource prior to submitting to the primary resource. Validation resources provide detailed error messages without modifying data. Primary resources fail immediately on detecting any error, and revert any changes that had previously been processed.
9. Build and test your application against the beta environment; once tested and debugged, switch your resource URLs to point to production.

Other getting started information

Case sensitivity

- All URLs, element names, and enumerated values in RelaxNG schemas are case-sensitive.

Resource misuse

- Watermark Faculty Success strives to build robust and tolerant software, however we cannot always anticipate how other developers may use our web services. Web services usage in excess of 10,000 requests per day per client is concerning, and may result in Watermark reaching out to pursue more efficient means of meeting the given need. It is rare, however, that clients reach this level of activity since batching of requests and caching of data can be used to diminish the number of calls required. It is also important to note that web services requests to Faculty Success for the purposes of integrating with another data source on campus should be made outside of business hours (7 am - 7 pm CT). If we determine that your organization's web service usage is causing a detriment to the system, we may temporarily disable your web service accounts until we are able to contact your Faculty Success Administrator.

Service accounts

- Schema Resource: Read Access - Metadata about your data collection screen configurations
- Data Resources:
  - ◊ Read Access - Retrieve data already stored by Faculty Success
  - ◊ Write Access - Import data into Faculty Success from another source, or update data already stored
  - ◊ Delete Access - Delete records stored by Faculty Success
- User Resources:
  - ◊ Read Access - View existing user accounts
  - ◊ Write Access - Create or modify user accounts and data collection access
- Role Resources:
  - ◊ Read Access - View details of security role configuration and user role assignments
  - ◊ Write Access - Add or remove user role assignments

Service glossary

Watermark Faculty Success' web services share common grammar when constructing resource URL.

{Date}

An ISO 8601 ("yyyy-MM-dd") formatted date. For example, "2007-10-13".

{DateQuery}

A URL query string that specifies a combination of "start" and/or "end" dates when calling the Data Query Resource. For example,"start=2007-10-13" includes records that occur after the specified date, while "end=2007-10-13" includes records that occur before. Specifying both, as "start=2007-10-13&end=2007-10-13", returns only records that occur on the specified date.

{EntityKey}

An identifier that represents a data collection entity, which is represented to end users as a single data collection page. Entity identifiers correspond with XML elements returned by the Data Query Resource. Available {EntityKey} values may be retrieved from the Schema Entity List Resource.

{EntityKeys}

A delimited collection of {EntityKey} strings. Entities are delimited by comma characters. Available {EntityKey} values may be retrieved from the Schema Entity List Resource.

{fullTextSearch}

A URL query string that specifies a search value to find matching records. For example, "fullTextSearch=%22mobile+phones%22" matches records containing the phrase "mobile phones." Searches analyze the entire record contents, including file attachments, and are not able to target specific fields.

{IndexEntryKey}

A value on an {IndexKey} axis that may be used to limit data retrieved. For example, the college index may allow values such as "Business", "Engineering", etc. Note that a single user may have multiple values for an index; i.e. a faculty member that belongs to both the Department of "Mathematics" and the Department of "Statistics". Available {IndexEntryKey} values may be retrieved from the Schema Index List Resource.

{IndexKey}

An axis that may be used to limit data retrieved. Common indexes include "COLLEGE", "DEPARTMENT", etc. Available {IndexKey} values may be retrieved from the Schema Index List Resource.

{IndexKeyEntryKeys}

A delimited collection of {IndexKey} and {IndexEntryKey} pairs. {IndexKeys} and Entry Keys are separated by colon characters, and multiple Key/Value pairs are delimited by commas characters. For example, to specify everyone from the Business College as well as everyone from the Mathematics Department, the {IndexKeyEntryKeys} string of "COLLEGE:Business,DEPARTMENT:Mathematics" would be used. Available {IndexKey} and {IndexEntryKey} values may be retrieved from the Schema Index List Resource.

{RoleKey}
Unique identifier for a security role. A security Role specifies a set of capabilities that may be applied to one or more users. Available {RoleKey} values may be retrieved from the Role List Resource.

{SchemaKey}
Unique identifier for a data source. Available {SchemaKey} values may be retrieved from the Schema List Resource.

{Username}
Unique identifier for a user. This is the identifier users enter to log into Faculty Success. This is often the user's email address without the domain name or a unique identifier assigned by your organization. Available {Username} values may be retrieved from the User List Resource.

{UserIdentifierType}
Unique identifier for a user, configured by Faculty Success for your organization. A user may have a unique user identifier value for a user identifier type, allowing you to integrate your user identities into our web services easily. Available {UserIdentifierType} values and associated user identifier values may be retrieved from the User List Resource.

{UserSchemaKey}
Unique identifier for a user's relationship to a data source. Available {UserSchemaKey} values for a User may be retrieved from the User Schema List Resource.

Accounts and privileges

A special service account is required for web service access; normal user accounts will not work. If you have not been provided service account credentials, please ask your Watermark Faculty Success Administrator to request that an account be created. Please indicate which service resources privileges you wish to use. Privileges can always be increased later, but should your service account ever be compromised, minimal privileges may decrease potential damage.

Service account privileges

- **Schema resource**: Read Access - Metadata about your data collection screen configurations
- **Data resources:**
  - ◊ Read Access - Retrieve data already stored by Faculty Success
  - ◊ Write Access - Import data into Faculty Success from another source, or update data already stored
  - ◊ Delete Access - Delete records stored by Faculty Success
- **User Resources:**
  - ◊ Read Access - View existing user accounts
  - ◊ Write Access - Create or modify user accounts and data collection access
- **Role Resources:**
  - ◊ Read Access - View details of security role configuration and user role assignments
  - ◊ Write Access - Add or remove user role assignments

A couple of notes regarding these resources:

- If you choose to include Read or Write access to the User Resource, we recommend also including Read or Write for the Role Resource, as accounts cannot be created fully without the ability to assign security roles.
- While Schema Resources is listed as an “option” above, it will automatically be included with each service account setup.

Base domain

Base URL

```
https://webservices.digitalmeasures.com/
```

Base URL in BETA for testing

```
https://betawebservices.digitalmeasures.com/
```

->> The beta database is refreshed from production each Saturday at 5:00 a.m. US/Central; any data that is changed in beta will be reverted when a refresh occurs

Resources by method

Methods

- **GET** is used for three primary purposes:
  - ◊ Pull data out of the system
  - ◊ Determine allowed values
  - ◊ Determine allowed formats for each resource
- **PUT** is used to update User related information:
  - ◊ Update a user's information that displays in Users and Security
  - ◊ Update a user's security assignments
  - ◊ Disable user account
- **POST** is used to create new user accounts and to create and update records and access settings

GET

GET is used for three primary purposes:

- Pull data out of the system
- Determine allowed values
- Determine allowed formats for each resource

Documentation – Provides access to posted Web Services documentation

```
GET login/service/v4/Documentation
```

Schema

A Schema is a representation of the set of data collection screens used by your organization. Your organization may have a single schema if you are working with Watermark Faculty Success at an institutional level. Your organization may have multiple schemas if you are using more than one data collection solution, or several units within your organization are working with Watermark Faculty Success separately.

Schema List Resource – Returns SchemaKey values (instruments)

```
GET login/service/v4/Schema
```

Schema List RelaxNG - Formally defines the document available via the Schema Entity List Resource.

```
GET /login/service/v4/Schema:list-relaxng
```

Schema entity

A Schema Entity corresponds to a single data collection screen.

Schema Entity List Resource – Returns list of activity screen names

```
GET login/service/v4/SchemaEntity/{SchemaKey}
```

example -> All screens for the university:

```
GET /login/service/v4/SchemaEntity/INDIVIDUAL-ACTIVITIES-University
```

Schema Entity List RelaxNG - Formally defines the document available via the Schema Entity List Resource

```
GET /login/service/v4/SchemaEntity:list-relaxng/{SchemaKey}
```

Schema Index

Schema Indexes are used to search records based on pre-defined criteria. Indexed data is primarily stored in one or more questions within a Schema. Default indexes include college, department, and username.

If the default indexes are not sufficient, please ask your Watermark Faculty Success Administrator to request that a new index be created. An index may be created on any data collection question, but are most useful when the question pertains to the user, not to an individual data record within the system.

Schema Index Resource – Returns list of valid indexes, such as Colleges & Departments

```
GET login/service/v4/SchemaIndex/{SchemaKey}
```

example -> All indexes and entries for the university:

```
GET /login/service/v4/SchemaIndex/INDIVIDUAL-ACTIVITIES-University
```

SchemaIndex with IndexKeyEntryKeys - Returns a subset list of valid indexes within an Index Entry

```
GET login/service/v4/SchemaIndex/{SchemaKey}/{IndexKeyEntryKeys}
```

example -> Indexes and entries for the College of Business:

```
GET/login/service/v4/SchemaIndex/INDIVIDUAL-ACTIVITIES-University/
COLLEGE:Business
```

example -> Indexes and entries for the College of Business and the College of Engineering:

```
GET/login/service/v4/SchemaIndex/INDIVIDUAL-ACTIVITIES-University/
COLLEGE:Business,COLLEGE:Engineering
```

Schema Index List RelaxNG - Formally defines the document available via the Schema Index List Resource.

```
GET /login/service/v4/SchemaIndex:list-relaxng/{SchemaKey}
```

User

User Resources provide read and write access to user accounts stored by Watermark Faculty Success.

User List Resource – Returns list of all users

```
GET login/service/v4/User
```

User List RelaxNG Resource - Formally defines the document available via the User List Resource.

```
GET /login/service/v4/User:list-relaxng
```

User/{SchemaKey} – Returns list of users for a specific Instrument

```
GET login/service/v4/User/{SchemaKey}
```

example -> Return list of users for the university

```
GET login/service/v4/User/INDIVIDUAL-ACTIVITIES-University
```

User/{SchemaKey}/{IndexKeyEntryKey} – Returns list of users for a specific Instrument and unit

```
GET login/service/v4/User/{SchemaKey}/{IndexKeyEntryKey}
```

example -> Return list of users for the College of Business

```
GET login/service/v4/User/INDIVIDUAL-ACTIVITIES-University/
COLLEGE:Business
```

example -> Return list of users for the departments of education and Medicine

```
GET login/service/v4/User/INDIVIDUAL-ACTIVITIES-University/
DEPARTMENT:Medicine,DEPARTMENT:Education
```

User Update RelaxNG Resource - Formally defines the document accepted by the User Update Resource.

```
GET/login/service/v4/User:update-relaxng
```

User Item Resource – Returns details of a single user's account

```
GET login/service/v4/User/USERNAME:{username}
```

User Item RelaxNG Resource - Formally defines the document available via the User Item Resource

```
GET /login/service/v4/User:item-relaxng
```

User schema

A User Schema resource links a User account to a set of data collection screens. A User Schema relationship must be established before any data may be stored for a user.

A User Schema resource also contains a partial representation of the user's data.

User Schema List Resource - Returns a list of schemas configured to the specified user

```
GET /login/service/v4/UserSchema/USERNAME:{Username}
```

User Schema List RelaxNG Resource - Formally defines the document available via the User Schema List Resource.

```
GET /login/service/v4/UserSchema:list-relaxng/
```

User Schema Item Resource - Presents the details of the specified user's relationship with a Schema, including {IndexKey} and {IndexEntryKey} pairs, and a partial representation of the user's data.

```
GET /login/service/v4/UserSchema/USERNAME:{Username}/{UserSchemaKey}
```

example -> Return details for user "MJordan"

```
GET login/service/v4/UserSchema/USERNAME:MJordan/INDIVIDUAL-
ACTIVITIES-University
```

User Schema Item RelaxNG Resource - Formally defines the document available via the User Schema Item Resource.

```
GET /login/service/v4/UserSchema:item-relaxng/
```

User Schema Create RelaxNG Resource - Formally defines the document accepted by the User Schema Create Resource.

```
GET /login/service/v4/UserSchema:create-relaxng
```

User Schema Update RelaxNG Resource - Formally defines the document accepted by the User Schema Update Resource.

```
Java
GET /login/service/v4/UserSchema:update-relaxng
```

Role

A role resource defines a set of security capabilities that may be assigned to one or more users.

Roles are composed of capabilities which grant access to a set of functionality related to a single Schema. Some capabilities allow additional levels of configuration.

Role List – Find all roles and available permissions

```
GET login/service/v4/Role
```

Role User List RelaxNG Resource - Formally defines the document available via the Role User List Resource.

```
GET /login/service/v4/RoleUser:list-relaxng
```

Role Item Resource – Find a list of user permissions (capabilities) by security role.

```
GET login/service/v4/Role/{roleKey}
```

example -> Return all permissions that are assigned to the faculty security role

```
GET login/service/v4/Role/INDIVIDUAL-ACTIVITIES-University-Faculty
```

example -> Return all permissions that are assigned to the college administrator security role

```
GET login/service/v4/Role/INDIVIDUAL-ACTIVITIES-University-
CollegeAdministrator
```

Role Item RelaxNG Resource - Formally defines the document available via the Role Item Resource

```
GET /login/service/v4/Role:item-relaxng[/{SchemaKey}]
```

Role User List Resource – Find a list of users by security role

```
GET login/service/v4/RoleUser/{roleKey}
```

example -> Return all users who are assigned to the college administrator security role

```
GET login/service/v4/RoleUser/INDIVIDUAL-ACTIVITIES-University-Colle
geAdministratorCollegeAdministrator
```

User role

A User Role resource links a user account to a security role to grant authorization and data access. A User Role resource may contain one or more permissions that restrict a user's access to data; the role's scope determines the availability of data permissions.

User Role List Resource – See security role and details for a specific user

```
GET login/service/v4/UserRole/USERNAME:{username}
```

User Role List RelaxNG Resource - Formally defines the document available via the User Role List Resource.

```
GET /login/service/v4/UserRole:list-relaxng
```

User Role Item Resource - Presents the details of the specified User Role relationship, including assigned permissions when available.

```
GET /login/service/v4/UserRole/USERNAME:{Username}/{RoleKey}
```

User Role Item RelaxNG Resource - Formally defines the document available via the User Role Item Resource.

```
GET /login/service/v4/UserRole:item-relaxng
```

User Role Create RelaxNG Resource - Formally defines the document accepted via the User Role Create Resource.

```
GET /login/service/v4/UserRole:create-relaxng
```

User Role Update RelaxNG Resource - Formally defines the document accepted via the User Role Update Resource.

```
GET /login/service/v4/UserRole:update-relaxng
```

Schema data

Schema Data represents data stored by Watermark Faculty Success for one or more users. These are the resources that will enable your institution to set up web profiles for your faculty by retrieving all their activity data that is to be displayed. You will be able to also narrow the focus of the data you are trying to pull by using additional parameters to determine what group of people you are querying, the date ranges these activities happen over, and any additional search values as well.

Schema Data Query resource - Retrieve data for a user or set of users by group

```
GET /login/service/v4/SchemaData/{SchemaKey}/{IndexKeyEntryKeys}
```

Schema Data Query resource - Retrieve data for a set of users by activity group

```
GET /login/service/v4/SchemaData/{SchemaKey}/{EntityKeys}
```

This resource is restricted to 20,000 Entity records per request. If you encounter a “Request too large” error, please narrow the scope of your request by using more narrow {IndexKeyEntryKeys}, specifying fewer {EntityKeys}, or specifying a smaller {DateQuery} or {fullTextSearch}

Schema Data Query resource parameter to retrieve data by a specific date range and (&) text search for value match

```
?{DateQuery}&{fullTextSearch}
```

*** Assumes Entity Keys are using the base configuration. Your Watermark Faculty Success administrator also has an access to a configuration report spreadsheet which will outline all activity codes and their field's codes as well. This resource may be helpful in understanding the type of data your institution is capturing data for and what specific type of data as well.

Faculty data and activities

{SchemaKey}/PCI - Retrieve all personal and contact information for all users in the university

```
GET /login/service/v4/SchemaData/{SchemaKey}/PCI
```

{SchemaKey}/SCHTEACH - Retrieve all courses taught for all users in the university

```
GET /login/service/v4/SchemaData/{SchemaKey}/SCHTEACH
```

{SchemaKey}/INTELLCONT - Retrieve all publications for all users in the university

```
GET /login/service/v4/SchemaData/{SchemaKey}/INTELLCONT
```

{SchemaKey}/PRESENT - Retrieve all presentations for all users in the university

```
GET /login/service/v4/SchemaData/{SchemaKey}/PRESENT
```

Faculty data and activities with additional parameters SAMPLES

{SchemaKey}/{IndexKeyEntryKeys}/PCI - Retrieve all personal and contact information for all users in the specific groups specified

```
GET /login/service/v4/SchemaData/{SchemaKey}/{IndexKeyEntryKeys}/PCI
```

example -> Retrieve all personal and contact information for all users in the Department of Mathematics and Department of Statistics

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-University/
DEPARTMENT: Mathematics,DEPARTMENT:Statistics/PCI
```

{SchemaKey}/{EntityKey}?{DateQuery} - Retrieve all activities for all users in the university in a specific date range

```
GET /login/service/v4/SchemaData/{SchemaKey}/{EntityKey}?{DateQuery}
```

example -> Retrieve all courses taught for all users in the university in the year of 2007

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-University/
SCHTEACH?startDate=2007-01-01&endDate=2007-12-31
```

example -> Retrieve all courses taught for all users in the department of Economics and Statistics in the year of 2007

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-University/
DEPARTMENT:Economics,DEPARTMENT:Statistics/SCHTEACH?startDate=2007-
01-01&endDate=2007-12-31
```

{SchemaKey}/{EntityKey} - Retrieve all activities for a specific user(s)

```
GET /login/service/v4/SchemaData/{SchemaKey}/USERNAME:{username}/
{EntityKey}
```

example -> Retrieve all publications for the specific user 'MJordan'

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-University/
USERNAME:MJordan/INTELLCONT
```

{SchemaKey}/{EntityKey}?{fullTextSearch} - Retrieve all activities that match a specific search value

```
GET /login/service/v4/SchemaData/{SchemaKey}/
{SchemaKey}?{fullTextSearch}
```

**** Searches analyze the entire record contents, including file attachments, and are not able to target specific fields.*

example -> Retrieve all presentations that contain the phrase 'mobile phones'

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-University/
PRESENT?fullTextSearch=%22mobile+phones%22
```

example -> Retrieve all presentations for all users in the College of Sciences in the year of 2021 that contain the phrase 'coronovirus'

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-
University/COLLEGE:College of Sciences/PRESENT?startDate=2021-01-
01&endDate=2021-12-31?fullTextSearch=%22coronavirus%22
```

Allow Sharing Parameter - Records that the user has chose to share for web profiles via the user interface through a yes/no toggle.

{SchemaKey}/{EntityKey}?include-=dmas - Retrieve all activities for users that have allowed their records to be shared (web profiles)

```
GET /login/service/v4/SchemaData/{SchemaKey}/{EntityKey}?include-=dmas
```

example -> Retrieve all publications for users that have allowed their records to be shared (web profiles)

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-University/
INTELLCONT?include-=dmas
```

example -> Retrieve all publications for users in the department of Economics that have allowed their records to be shared (web profiles)

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-University/
DEPARTMENT:Economics/INTELLCONT?include-=dmas
```

example -> Retrieve all research in progress for users in the department of Mathematics for the year of 2022 that have allowed their records to be shared (web profiles)

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-University/
DEPARTMENT:Mathematics/RESPROG?startDate=2021-01-01&endDate=2021-12-
31??include-=dmas
```

example -> Retrieve all research in progress for the user 'MJordan' for the year of 2022 that have allowed their records to be shared (web profiles)

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-University/
USERNAME:MJordan/RESPROG?startDate=2021-01-01&endDate=2021-12-
31??include-=dmas
```

Schema Data RelaxNG Resource - Formally defines the document available via the Schema Data Query Resource and accepted by the Schema Data Import Resource.

```
GET /login/service/v4/SchemaData:relaxng/{SchemaKey}
```

Schema Data Delete RelaxNG Resource - Formally defines the document accepted by the Schema Data Delete Resource

```
GET /login/service/v4/SchemaData:delete-relaxng/{SchemaKey}
```

Schema data backup

Schema Data Backup Resource - Download a complete backup of your Schema Data, refreshed on a weekly basis. Backups are returned as a Zip file containing a CSV data for each Entity in the system, suitable for importing into a relational database system.

```
GET /login/service/v4/SchemaData:backup/{SchemaKey}
```

Backups are not enabled by default; if you require weekly backups, please ask your Watermark Faculty Success Administrator to submit a task request to enable this functionality.

POST

POST is used to create new user accounts and to create and update records and access settings

User

User Create Resource

example -> create Fred Flintstone with username “FFlintstone”

```
POST /login/service/v4/User
```

Body

```
<User username=”FFlintstone”>
   <FirstName>Frederick</FirstName>
   <MiddleName>J</MiddleName>
   <LastName>Flintstone</LastName>
   <Email>fflintstone@bedrock.edu</Email
   <LocalAuthentication>wilma123</LocalAuthentication>
</User>
```

Create multiple users

example -> Fred Flintstone with username “FFlintstone” and Josh Smith with username "jsmith"

Body

```
<User username=”FFlintstone”>
   <FirstName>Frederick</FirstName>
   <MiddleName>J</MiddleName>
   <LastName>Flintstone</LastName>
   <Email>fflintstone@bedrock.edu</Email
</User>
<User username=”jsmith”>
   <FirstName>Josh</FirstName>
   <MiddleName>D</MiddleName>
   <LastName>Smith</LastName>
   <Email>jsmith@bedrock.edu</Email
</User>
```

Response -> Success

```
<dmu:Successxmlns:xlink=”http://www.w3.org/1999/xlink”>
   <Updated xlink:type=”simple” xlink:href=”/login/service/v4/User/
USERNAME:FFlintstone”/>
</dmu:Success>
```

User identifier values may be used in addition to the username attribute if your organization has set up user identifier types through Faculty Success. This allows you to provide additional custom unique user identifying values for a user. For example, if your organization has a {UserIdentifierType} of “bannerId”, you may specify the user using the @bannerId attribute:

```
<User username=”FFlintstone” bannerId=”323”>
...
</User>
```

User Create RelaxNG Resource - Formally defines the document accepted by the User Create Resource.

```
GET /login/service/v4/User:create-relaxng
```

User Create Validation Resource - Perform a “dry run” on a user to be created.

```
POST /login/service/v4/User:create-validate
```

This resource is nearly identical to the User Create Resource, except for the following differences:

- Data are not initially validated against the RelaxNG schema produced by the User Create RelaxNG Resource. Instead, a more forgiving interval validation mechanism is used that categorizes and reports all errors found, instead of a single error per request.
- All processing steps are performed except for saving the user changes.
- If no other errors are found, the requested data is validated against the RelaxNG schema as a final check.

User Schema

User Schema Create Resource - Create a new User Schema relationship, using the same format as returned by the User Schema Item Resource, and formally defined by the User Schema Create RelaxNG Resource.

```
POST /login/service/v4/UserSchema/USERNAME:{Username}
```


example -> Create User Schema Relationship , Grant Barney Rubble access to the University data collection screens

```
POST /login/service/v4/UserSchema/USERNAME:BRubble
```

Body

```
<INDIVIDUAL-ACTIVITIES-University>
   <ADMIN>
        <AC_YEAR>2007-2008</AC_YEAR>
        <ADMIN_DEP>
            <DEP>Management</DEP>
        </ADMIN_DEP>
   </ADMIN>
</INDIVIDUAL-ACTIVITIES-University>
```

User Schema Create Validation Resource - Perform a “dry run” on user schema relationship to be created.

```
POST /login/service/v4/UserSchema:create-validate/
```

This resource is nearly identical to the User Schema Create Resource, except for the following differences:

- Data are not initially validated against the RelaxNG schema produced by the User Create RelaxNG Resource. Instead, a more forgiving interval validation mechanism is used that categorizes and reports all errors found, instead of a single error per request.
- All processing steps are performed except for saving the user schema changes.
- If no other errors are found, the requested data is validated against the RelaxNG schema as a final check.

User Schema Update Validation Resource - Perform a “dry run” on a user schema relationship to be updated.

```
POST /login/service/v4/UserSchema:update-validate
```

This resource is nearly identical to the User Schema Update Resource, except for the following differences:

- Data are not initially validated against the RelaxNG schema produced by the User Create RelaxNG Resource. Instead, a more forgiving interval validation mechanism is used that categorizes and reports all errors found, instead of a single error per request.
- All processing steps are performed except for saving the user schema changes.
- If no other errors are found, the requested data is validated against the RelaxNG schema as a final check.

User role

User Role Create Resource - Create a new User Role relationship, using the same format as returned by the User Role Item Resource, and formally defined by the User Role Create RelaxNG Resource.

```
POST /login/service/v4/UserRole/USERNAME:{Username}
```

example ->> Create User Role Relationship, Grant Fred Flintstone the Department Chair role for the Economics department

```
POST /login/service/v4/UserRole/USERNAME:FFlintstone
```

Body

```
<INDIVIDUAL-ACTIVITIES-Business-DEPARTMENTCHAIR>
   <Permission permissionKey="Economics"/>
</INDIVIDUAL-ACTIVITIES-Business-DEPARTMENTCHAIR>
```

User Role Create Validate Resource - Perform a "dry run" on user role relationship to be created.

```
POST /login/service/v4/UserRole:create-validate/USERNAME:{Username}
```

This resource is nearly identical to the User Role Create Resource, except for the following differences:

- Data are not initially validated against the RelaxNG schema produced by the User Role Create RelaxNG Resource. Instead, a more forgiving interval validation mechanism is used that categorizes and reports all errors found, instead of a single error per request.
- All processing steps are performed except for saving the user schema changes.
- If no other errors are found, the requested data is validated against the RelaxNG schema as a final check.

Schema data

Schema Data Import Resource - Import data for one or more users, using the same format returned from the Schema Data Query Resource as formally defined by the Schema Data RelaxNG Resource.

```
POST /login/service/v4/SchemaData/{SchemaKey}
[/{IndexKeyEntryKeys}]
[/{EntityKeys}]
```

Import data for a user or set of users by group

```
POST /login/service/v4/SchemaData/{SchemaKey}/{IndexKeyEntryKeys}
```

example -> Insert two new Intellectual Contribution records into Fred Flintstone's data

Post

```
/login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-Business/
USERNAME:FFlintstone
```

Body

```
<Data>
   <Record username="FFlintstone">
        <INTELLCONT>
            <CONTYPE>Book, Edited</CONTYPE>
            <TITLE>Brontosaurus Crane Service Manual</TITLE>
            <!-- ... -->
        </INTELLCONT>
        <INTELLCONT>
            <CONTYPE>Journal Article</CONTYPE>
            <TITLE>Optimization of Brontosaurus Based
MiningOperations</TITLE>
            <!-- ... -->
        </INTELLCONT>
   </Record>
</Data>
```

Update Records by Id - Update Fred Flintstone's and Wilma Flintstone's Scheduled Teaching - Mean Course Evaluation Score based on results from the Query Resource

Current Data for Fred Flintone's existing administrative data

```
GET /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-Business/SCHTEACH
```

Response

```
<Dataxmlns:dmd="http://www.digitalmeasures.com/schema/data-
metadata"dmd:date="2007-09-27">
   <Record userId="12345" username="FFlintstone" dmd:id="10002">
        <SCHTEACH id="21235" dmd:lastModified="2007-03-23T15:23:56"
            <MEAN_EVAL>3.25</MEAN_EVAL>
        </SCHTEACH>
   </Record>
   <Record userId="56232" username="wflintstone" dmd:id="10352"
        <SCHTEACH id="62345" dmd:lastModified="2007-03-23T15:28:39"
            <MEAN_EVAL>3.45</MEAN_EVAL>
        </SCHTEACH>
   </Record>
</Data>
```

Update Fred Flintstone's and Wilma Flintstone's Scheduled Teaching - Mean Course Evaluation Score based on results from the Query Resource

```
POST /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-Business
```

Body

```
<Dataxmlns:dmd="http://www.digitalmeasures.com/schema/data-
metadata"dmd:date="2007-09-27">
   <Record userId="12345" username="FFlintstone"
        <SCHTEACH id="21235"
            <MEAN_EVAL>5.75</MEAN_EVAL>
        </SCHTEACH>
   </Record>
   <Record userId="56232" username="wflintstone"
        <SCHTEACH id="62345"
            <MEAN_EVAL>6.75</MEAN_EVAL>
        </SCHTEACH>
   </Record>
</Data>
```

Update Records by Primary Key - Update Fred Flintstone's and Wilma Flintstone's Scheduled Teaching - Mean Course Evaluation Score based on a Primary Key of Year, Term, Course Prefix, Number and Section

```
POST /login/service/v4/SchemaData/INDIVIDUAL-ACTIVITIES-Business
```

Body

```
<Data>
   <Record username="FFlintstone">
        <SCHTEACH>
            <TYY_TERM>2007-2008</TYY_TERM>
            <TYT_TERM>Fall</TYT_TERM>
            <COURSEPRE>ACCT</COURSEPRE>
            <COURSENUM>101</COURSENUM>
            <SECTION>1</SECTION>
            <MEAN_EVAL>3.25</MEAN_EVAL>
        </SCHTEACH>
   </Record>
   <Record username="wflintstone"
        <SCHTEACH>
            <TYY_TERM>2007-2008</TYY_TERM>
            <TYT_TERM>Fall</TYT_TERM>
            <COURSEPRE>ACCT</COURSEPRE>
            <COURSENUM>302</COURSENUM>
            <SECTION>8</SECTION>
            <MEAN_EVAL>3.45</MEAN_EVAL>
        </SCHTEACH>
   </Record>
</Data>
```

Schema Data Validate Resource - Perform a “dry run” on data to be imported.

```
POST /login/service/v4/SchemaData:validate/{SchemaKey}
[/{IndexKeyEntryKeys}]
[/{EntityKeys}]
```

This resource is nearly identical to Schema Data Import Resource, except for the following differences:

- Data are not initially validated against the RelaxNG schema produced by the Schema Data RelaxNG Resource. Instead, a more forgiving interval validation mechanism is used that categorizes and reports all errors found, instead of a single error per request.

If no other errors are found, the requested data is validated against the RelaxNG schema as a final check.

Schema Data Delete Resource - Permanently delete data records for one or more users, using the format formally defined by the Schema Data Delete RelaxNG Resource.

```
POST/login/service/v4/SchemaData:delete/{SchemaKey}/
[{IndexKeyEntryKeys}][?{DateQuery}]
```

Schema Data Delete Validate Resource - Perform a “dry run” on data to be deleted. This resource is nearly identical to Schema Data Delete Resource, except for the data is not deleted.

```
POST /login/service/v4/SchemaData:delete-validate/{SchemaKey}/
[{IndexKeyEntryKeys}][?{DateQuery}]
```

example -> Delete Entity Data - Delete all Scheduled Teaching records.

```
POST /login/service/v4/SchemaData:delete/INDIVIDUAL-ACTIVITIES-
University
```


Body

```
<Data>
<SCHTEACH>
<all/>
</SCHTEACH>
</Data>
```

example -> Delete Entity Data by IDs - Delete all Scheduled Teaching records by ID

```
POST /login/service/v4/SchemaData:delete/INDIVIDUAL-ACTIVITIES-
University
```

Body

```
<Data>
<SCHTEACH>
<itemid="298357987"/>
<itemid="298357235"/>
</SCHTEACH>
</Data>
```

example ->Delete User's Entity Data - Delete all of Fred Flintstone's Scheduled Teaching records.

```
POST /login/service/v4/SchemaData:delete/INDIVIDUAL-ACTIVITIES-
University/USERNAME:FFlintstone
```

Body

```
<Data>
<SCHTEACH>
<all/>
</SCHTEACH>
</Data>
```

Schema Data Delete Validate Resource - Perform a “dry run” on data to be deleted. This resource is nearly identical to Schema Data Delete Resource, except for the data is not deleted.

```
POST /login/service/v4/SchemaData:delete-validate/{SchemaKey}/
[{IndexKeyEntryKeys}][?{DateQuery}]
```

User batch resources

User Batch resources provide a consolidated tool for creating and updating multiple user accounts, user schema relationships, and user role relationships. The resource automatically determines if a specified resource should be created or updated based on the presence of existing records.

The batch resources are a convenient way to set up multiple user accounts and privileges in a single request, and are best suited for setting up a number of new users at the beginning of a term or year. The non-batch user resources are more appropriate for day-to-day management of individual users and privileges.

User Batch Save Resource - Creates or updates one or more users using the same format returned from the List Resource, as formally defined by the User Batch RelaxNG Resource.

```
POST /login/service/v4/UserBatch[/{SchemaKey}]
```

example → Create two new users with Yearly Data records

Body

```
<Users>
<User username="sampleuser1" enabled="true">
 <LocalAuthentication>test</LocalAuthentication>
 <FirstName>Sample</FirstName>
 <MiddleName></MiddleName>
 <LastName>User</LastName>
 <Email>sampleuser1@watermarkinsights.com</Email>
 <UserRoles>
  <INDIVIDUAL-ACTIVITIES-University-Faculty/>
 </UserRoles>
 <UserSchemas>
  <INDIVIDUAL-ACTIVITIES-University>
   <ADMIN>
```

```
    <AC_YEAR>2022-2023</AC_YEAR>
    <COLLEGE>College of Law</COLLEGE>
    <ADMIN_DEP>
     <DEP>Law</DEP>
     <DISCIPLINE>Business Law</DISCIPLINE>
     <SPECIALTY>Management</SPECIALTY>
    </ADMIN_DEP>
   </ADMIN>
  </INDIVIDUAL-ACTIVITIES-University>
 </UserSchemas>
</User>
<User username="sampleuser2" enabled="true">
 <LocalAuthentication>test</LocalAuthentication>
 <FirstName>Sample</FirstName>
 <MiddleName></MiddleName>
 <LastName>User</LastName>
 <Email>sampleuser2@watermarkinsights.com</Email>
 <UserRoles>
  <INDIVIDUAL-ACTIVITIES-University-Faculty/>
 </UserRoles>
 <UserSchemas>
  <INDIVIDUAL-ACTIVITIES-University>
   <ADMIN>
    <AC_YEAR>2022-2023</AC_YEAR>
    <COLLEGE>College of Law</COLLEGE>
    <ADMIN_DEP>
     <DEP>Law</DEP>
     <DISCIPLINE>Business Law</DISCIPLINE>
     <SPECIALTY>Management</SPECIALTY>
    </ADMIN_DEP>
   </ADMIN>
  </INDIVIDUAL-ACTIVITIES-University>
 </UserSchemas>
</User>
</Users>
```


Bulk update roles

example → add the “Department: Law” security role to 2 users

Body

```
<Users>
   <User username="sampleuser1">
       <UserRoles>
           <INDIVIDUAL-ACTIVITIES-University-Department>
               <Permission permissionKey='Law'/>
           </INDIVIDUAL-ACTIVITIES-University-Department>
       </UserRoles>
   </User>
   <User username="sampleuser2">
       <UserRoles>
           <INDIVIDUAL-ACTIVITIES-University-Department>
               <Permission permissionKey='Law'/>
           </INDIVIDUAL-ACTIVITIES-University-Department>
       </UserRoles>
   </User>
</Users>
```

Bulk update user attribute

example → Update the email for the 2 users we created

Body

```
<Users>

<User username="sampleuser1"><Email>sampleuser1@testu.edu</Email></
User>

<User username="sampleuser2"><Email>sampleuser2@testu.edu</Email></
User>

</Users>
```

PUT

User

User Update Resource - Update an existing user account, using the same format as returned by the User Item Resource, and formally defined by the User Update RelaxNG Resource.

```
PUT /login/service/v4/User/USERNAME:{Username}
```

example -> Update User Details
Update Wilma Slaghoople's last name, email address, and username:

```
PUT /login/service/v4/User/USERNAME:WSlaghoople
```

Body

```
<User username="WFlintstone">
   <LastName>Flintstone</LastName>
   <Email>wflintstone@bedrock.edu</Email
</User>
```

Response

```
<dmu:Successxmlns:xlink="http://www.w3.org/1999/xlink">
   <Updated xlink:type="simple" xlink:href="/login/service/v4/User/
USERNAME:WFlintstone"/>
</dmu:Success>
```


example -> Update Custom User Identifer
Update Barney Rubble's bannerId:

```
PUT /login/service/v4/User/USERNAME:BRubble
```

Body

```
<User bannerId="999"/>
```

Response

```
<dmu:Successxmlns:xlink="http://www.w3.org/1999/xlink">

   <Updated xlink:type="simple" xlink:href="/login/service/v4/User/

USERNAME:BRubble"/>

</dmu:Success>
example -> Disable User Account
Disable Barney Rubble's account:
PUT /login/service/v4/User/USERNAME:BRubble
```

Body

```
<User enabled="false"/>
```

Response

```
<dmu:Successxmlns:xlink="http://www.w3.org/1999/xlink">

   <Updated xlink:type="simple" xlink:href="/login/service/v4/User/

USERNAME:BRubble"/>

</dmu:Success>
```


User Update Validation Resource - Perform a “dry run” on a user to be updated

```
PUT /login/service/v4/User:update-validate/USERNAME:{Username}
```

This resource is nearly identical to the User Update Resource, except for the following differences:

- Data are not initially validated against the RelaxNG schema produced by the User Update RelaxNG Resource. Instead, a more forgiving interval validation mechanism is used that categorizes and reports all errors found, instead of a single error per request.
- All processing steps are performed except for saving the user changes.
- If no other errors are found, the requested data is validated against the RelaxNG schema as a final check.

User Role

User Role Update Resource - Update a new User Role relationship, altering the Permissions assigned to a user, using the same format as returned by the User Role Item Resource, and formally defined by the User Role Update RelaxNG Resource.

```
PUT /login/service/v4/UserRole/USERNAME:{Username}/{RoleKey}
```

example -> Modify User Role Relationship, change Fred Flintstone’s permission for the Department Chair role to Management

```
PUT /login/service/v4/UserRole/USERNAME:Fflintston/INDIVIDUAL-
ACTIVITIES-Business
```

Body

```
<INDIVIDUAL-ACTIVITIES-Business-DEPARTMENTCHAIR>
   <Permission permissionKey=”Management”/>
</INDIVIDUAL-ACTIVITIES-Business-DEPARTMENTCHAIR>
```

User Role Update Validate Resource - Perform a “dry run” on user role relationship to be updated.

```
PUT /login/service/v4/UserRole:update-validate/USERNAME:{Username}/
{RoleKey}
```

This resource is nearly identical to the User Role Update Resource, except for the following differences:

- Data are not initially validated against the RelaxNG schema produced by the User Role Update RelaxNG Resource. Instead, a more forgiving interval validation mechanism is used that categorizes and reports all errors found, instead of a single error per request.
- All processing steps are performed except for saving the user schema changes.
- If no other errors are found, the requested data is validated against the RelaxNG schemaas a final check.

User Schema

User Schema Update Resource - Update an existing User Schema relationship, using the same format as returned by the User Schema Item Resource, and formally defined by the User Schema Update RelaxNG Resource.

```
PUT /login/service/v4/UserSchema/USERNAME:{Username}/{UserSchemaKey}
```

example -> Update Index Entries for User, Change Fred Flintstone’s departments to Management and Economics for 2007-2008t

Current Data for Fred Flintone's existing administrative data

```
GET /login/service/v4/UserSchema/USERNAME:Fflintston/INDIVIDUAL-
ACTIVITIES-University
```

Response

```
<INDIVIDUAL-ACTIVITIES-University>
   <dmd:IndexEntry indexKey="DEPARTMENT" entryKey="Management"
text="Management"/>
   <dmd:IndexEntry indexKey="DEPARTMENT" entryKey="Marketing"
text="Marketing"/>
   <ADMIN>
        <AC_YEAR>2007-2008</AC_YEAR>
        <ADMIN_DEP>
            <DEP>Management</DEP>
        </ADMIN_DEP>
        <ADMIN_DEP>
            <DEP>Marketing</DEP>
        </ADMIN_DEP>
   </ADMIN>
</INDIVIDUAL-ACTIVITIES-University>
```

Change Fred Flintstone's departments to Management and Economics for 2007-2008

```
PUT /login/service/v4/UserSchema/USERNAME:Fflintston/INDIVIDUAL-
ACTIVITIES-University
```

Body

```
<INDIVIDUAL-ACTIVITIES-University>
   <ADMIN>
        <AC_YEAR>2007-2008</AC_YEAR>
        <ADMIN_DEP>
            <DEP>Management</DEP>
        </ADMIN_DEP>
        <ADMIN_DEP>
            <DEP>Economics</DEP>
        </ADMIN_DEP>
   </ADMIN>
</INDIVIDUAL-ACTIVITIES-University>
```

DELETE

User

User Delete Resource - Permanently delete a user and all associated data.

```
DELETE /login/service/v4/User/USERNAME:{Username}
```

User Schema Delete Resource - Permanently delete a user's relationship with a schema and all associated data.

```
DELETE /login/service/v4/UserSchema/USERNAME:{Username}/
{UserSchemaKey}
```

User Role Delete Resource - Permanently delete a user's relationship with a security role.

```
DELETE /login/service/v4/UserRole/USERNAME:{Username}/{RoleKey}
```

Concept samples

Example: Faculty directory website

1. Retrieve a list of colleges and/or departments as stored by Faculty Success from the Schema Index Resource.

```
GET /login/service/v4/SchemaIndex/{SchemaKey}
```

1. Transform the returned index/entry lists into HTML, and include a link to a user list page for each College or Department, including the {IndexKey} and {EntryKey} in the URL or query string.
2. Retrieve a list of users for a college or department from the User List Resource.

```
GET /login/service/v4/User/{SchemaKey}/{IndexKey}:{EntryKey}
```

1. Transform the returned user list into HTML, and include a link to a user detail page for each User, including the {Username} in the URL or query string.
2. Retrieve user data for an individual user from the Schema Data Query Resource, requesting data from as many data collection screens as needed. {EntityKey} values for data collection screens may be retrieved from the Schema Entity Resource, and cached or hard-coded.

```
GET /login/service/v4/SchemaData/{SchemaKey}/ USERNAME:{Username}/
{EntityKeys}
```

1. Transform the returned user data into HTML, optionally integrating data from other university information systems.
2. Optional: Provide an HTML form to search for users by partial first and/or last name. Users can be searched via the User List Resource.

```
GET /login/service/v4/User/{SchemaKey}?firstName=F&lastName=Flintston
```

Example: Centralized account creation

1. For reference, retrieve a list of users currently stored in the system from the User List Resource.

```
GET /login/service/v4/User/
```

1. Create a new user by sending a correctly formatted XML request for user creation to the User Create Resource.

```
POST /login/service/v4/User/
```

Body

```
<User username="{Username}">
   <FirstName>{FirstName}</FirstName>
   <MiddleName>{MiddleName}</MiddleName>
   <LastName>{LastName}</LastName>
   <Email>{EmailAddress}</Email>
   <LocalAuthentication>{Password}</LocalAuthentication>
</User>
```

1. Grant the new user access to a Schema by sending a correctly formatted XML request for user creation to the User Schema Create Resource. Note: The exact XML required will vary based on your data collection screen configuration; consult the User Schema Create RelaxNG Resource for full details.

```
POST /login/service/v4/UserSchema/USERNAME:{Username}
```

Body

```
<{SchemaKey}>
   <ADMIN>
       <AC_YEAR>{AcademicYear}</AC_YEAR>
       <ADMIN_DEP>
           <DEP>{Department}</DEP>
       </ADMIN_DEP>
   </ADMIN>
</{SchemaKey}>
```

1. Grant the new user access to a security role by sending a correctly formatted XML request for user creation to the User Role Create Resource. Note: Not all security roles require a Permission entity; consult the Role List Resource or the User Role Create RelaxNG Resource for full details.

```
POST /login/service/v4/UserRole/USERNAME:{Username}
```

Body

```
<{RoleKey}>
<Permission permissionKey="{PermissionKey}"/>
</{RoleKey}>
```

Example: Automated data import

1. For reference, or when updating data, retrieve existing data using the Schema Data Query Resource.

```
GET /login/service/v4/SchemaData/{SchemaKey}
```

1. Transform your existing data into the format specified by the Schema Data RelaxNG Resource, which describes the same format returned by the Schema Data Query Resource.
2. Send the data for import to the Schema Data Validate Resource.

```
POST /login/service/v4/SchemaData:validate/{SchemaKey}
```

Body

```
<Data>
   <Record username="{Username1}">
       <PCI>
           <OPHONE1>{AreaCode1}</OPHONE1>
           <OPHONE2>{Exchange1}</OPHONE2>
           <OPHONE3>{Line1}</OPHONE3>
        </PCI>
   </Record>
<Record username="{Username2}">
        <PCI>
           <OPHONE1>{AreaCode2}</OPHONE1>
           <OPHONE2>{Exchange2}</OPHONE2>
           <OPHONE3>{Line2}</OPHONE3>
        </PCI>
   </Record>
</Data>
```


1. Resolve any validation errors reported, by correcting invalid values, or by asking your Watermark Faculty Success Administrator to request screen revisions necessary to accommodate your data.
2. Send the data for import to the Schema Data Import Resource.

```
POST /login/service/v4/SchemaData/{SchemaKey}
```

Body

```
<Data>
   <Record username="{Username1}">
       <PCI>
           <OPHONE1>{AreaCode1}</OPHONE1>
           <OPHONE2>{Exchange1}</OPHONE2>
           <OPHONE3>{Line1}</OPHONE3>
        </PCI>
   </Record>
<Record username="{Username2}">
        <PCI>
           <OPHONE1>{AreaCode2}</OPHONE1>
           <OPHONE2>{Exchange2}</OPHONE2>
           <OPHONE3>{Line2}</OPHONE3>
        </PCI>
   </Record>
</Data>
```

Example: Security role management

1. Retrieve a list of users currently stored in the system from the User List Resource.

```
GET /login/service/v4/User/
```


1. Transform the returned user list into HTML, and include a link to a user detail page for each user, including the {Username} in the URL or query string.
2. Retrieve existing security role assignments for a user from the User Role List Resource.

```
GET /login/service/v4/UserRole/USERNAME:{Username}
```

1. Transform the returned user roles into HTML, and include a link to a user role editing for each College or Department, including the {Username} and {RoleKey} in the URL or query string.
2. Retrieve existing security role assignment details for a user from the User Role Item Resource.

```
GET /login/service/v4/UserRole/USERNAME:{Username}/{RoleKey}
```

1. Retrieve the possible permissions for the security role from the Role Item Resource.

```
GET /login/service/v4/Role/{RoleKey}
```

1. Transform the returned user role details into an HTML form, and include list of possible permissions for the security role forediting.
2. Update the existing security role assignments with the selected permissions using the User Role Update Resource.

```
PUT /login/service/v4/UserRole/USERNAME:{Username}/{RoleKey}
```

1. To create a new association between a user and a security role, call the User Role Create Resource.
2. To delete a user's existing association with a security role, call the User Role Delete Resource.

Code samples

https://support.watermarkinsights.com/hc/en-us/sections/4409239751451-Sample-Code

Watermark gives higher education institutions the tools they need to easily track, manage, and examine their data. For over twenty years, colleges and universities have used Watermark solutions to complete assessment and accreditation requirements, capture and analyze student feedback, showcase faculty accomplishments, and improve student engagement. Watermark's Educational Impact Suite (EIS) puts data into context so faculty and staff can focus on what really matters: institutional and student success.

Learn why Watermark is trusted by over 1,700 colleges and universities to support continuous improvement at www.watermarkinsights.com.