Moody's Analytics Scenario Studio API User Guide

version 2

Introduction

Scenario Studio API provides easy programmatic control over Scenario Studio enabling process automation and workflow integration. We provide a Swagger API interface to provide additional documentation and facilitate testing. The Scenario Studio API supports both HMAC and oAuth 2.0 authentication and JSON responses, and is agnostic regarding the client's operating system and programming language. The API is throttled (rate-limited) to 300 requests per minute and one gigabyte of data per month. The Swagger documentation examples express the requests in cURL notation, and the GitHub repository contains examples in Python and R.

Overview

The Scenario Studio v2 API core functions consist of several groups of endpoints for creating, editing, managing, and reading data and metadata from Scenario Studio projects.

- /audit/... - Retrieve content and counts of records in a project's audit log
- /project/... - Get metadata for a project, add and remove scenarios, and configure project settings
    - /project/.../scenario/... - Get metadata for a scenario, run solves, and configure settings
    - /project/.../data-series/... - Retrieve and push time series data to and from Scenario Studio
    - /project/.../scenario/.../series/... - Get series information, claim/release, change state, and customize equations
    - /project/.../search/ - Returns lists and counts of series within a project according to search parameters
- .../order/... - Get status of brokerized processes such as solves and project builds
- /base-scenario/... - Search and retrieve information about available base scenarios

Authentication

The Scenarion Studio API supports 2 forms of authentication:

1. HMAC Signature
2. OAuth 2.0 Token

Both methods require Scenario Studio API access key and encryption key. Every request to the API must contain either an HMAC signature or OAuth Token.

Getting API Keys

Access to the API is controlled by the combination of an access key and an encryption key. These keys are issued to a single user. To obtain your keys, go to the "My Subscriptions" section of your Economy.com account: https://www.economy.com/myeconomy/api-key-info.

Figure 1. Example access key and encryption key

```
DB73FDF0-043C-4018-A7EB-CFB57356BA22
7C7C2FEA-6D18-49A1-BEC9-193B67EAE87D
```

Using HMAC Authentication

Authenticating each request

HMAC signature is generated from your access key, encryption key, and a time stamp. You must attach a signature to every request using HAMC authorization, and you must re-create the signature with every request; you will receive an HTTP 401 Unauthorized error otherwise.

The access key, time stamp and signature need to be passed in as HTTP headers (not as part of the query string). Do not transmit the encryption key in the request since it is a secret between you and the server. Specifically, the signature is a SHA256 hash of the access key, encryption key and time stamp. The time stamp must be formatted as yyyy-MM-ddTHH:mm:ssZ using UTC. For example, "July 30, 2018 5:03:28pm EST" must be represented as 2018-07-30T21:03:28Z.

Figure 2. Example HTTP request header

```
AccessKeyId: DB73FDF0-043C-4018-A7EB-CFB57356BA22
TimeStamp: 2012-08-02T14:25:20Z
Signature: A7808C5A67C422054364F195B16175308317930848232C6A08A77224F1017E83
```

Figure 3. Example signature creation in C

This C# function creates a signature from an access key, encryption key, and time stamp. See the GitHub repository for samples using Python and R.

```c# using System; using System.Text; using System.Security.Cryptography; public static string CreateHMACSignature (string accKey, string encKey, string timeStamp) { string signature = string.Empty; byte[] keyBytes = Encoding.UTF8.GetBytes(encKey); using (HMAC hmac = new HMACSHA256(keyBytes)) { byte[] bytesToHash = Encoding.UTF8.GetBytes(accKey + timeStamp); byte[] hashedBytes = hmac.ComputeHash(bytesToHash); for (int i = 0; i < hashedBytes.Length; i++) { signature += hashedBytes[i].ToString("X2"); } } return signature; }
```

```
## Using OAuth authentication.

oAuth Token can be generated by calling an API endpoint, using API access key as *client_id* and API encryption key as *client_secre

### Obtaining OAuth Token

The _oauth2/token_ endpoint is used to generate oAuth Token using your *access key as client_id, encryption key as client_secret and

##### Figure 4. Request
```

curl -X POST \ https://api.economy.com/scenario-studio/v1/oauth2/token \ -H 'Content-Type: application/x-www-form-urlencoded' \ -d 'client_id=DB73FDF0-043C-4018-A7EB-CFB57356BA22' \ -d 'client_secret=47C7C2FEA-6D18-49A1-BEC9-193B67EAE87D' \ -d 'grant_type=client_credentials'

```
The response to the above request will have a new access token.

##### Figure 5. Respone
```

{ "token_type": "bearer", "access_token": "SrZ5UkbzPn432zqMLgV3Ja", "expires_in": 3600 }

```
A request to API will have **Authorization: Bearer _token_** as header

##### Figure 6. Call to API endpoint using oAuth Token
```

curl -X GET \ 'https://api.economy.com/scenario-studio/v1/project' \ -H 'Authorization: Bearer SrZ5UkbzPn432zqMLgV3Ja' ```

Frequently asked questions

- General API use
  - What programming languages does the API support?
  - What response types are supported?
  - Can I use the API from Linux?
  - What kind of authentication does the API use?
  - How often do I need to regenerate the signature?
  - How often do I need to regenerate the token?
  - Is the API throttled?
  - Can I use the API to populate a data warehouse?
- API capabilities
  - What kind of Scenario Studio information can I retrieve?
  - Can I create or alter a project or scenario?
- Scenenario Studio API parameters
  - How should date parameters be constructed?
  - How many series can I retrieve with the data-series endpoint?
  - If I alter the name of a project using the Scenario Studio web application, do I need to change my code?
- Support
  - Whom do I contact for assistance in using the API?
  - Do other Moody's Analytics products have APIs?
  - I don't understand this jargon—can you translate?

What programming languages does the API support?

The programming language used at your end is immaterial, so long as it (a) creates HTTP requests that the API can process, and (b) can interpret the JSON-formatted responses produced by the API. The examples provided in this document and in the GitHub repository use cURL, Python and R.

What response types are supported?

JSON is the only response type returned by the API.

Can I use the API from Linux?

Yes, because the operating system is immaterial. Java, Python and R are commonly used on Linux machines; to run C#, you will need to install the .NET Core framework. Setting up your run-time environment is beyond the scope of this document.

What kind of authentication does the API use?

Our API uses HMAC and OAuth 2.0 authentication. See the Authentication section above for more info.

How often do I need to regenerate the HMAC signature?

You must re-create the signature prior to every request; otherwise you will receive the "HTTP 401 Unauthorized" error. You may find it useful to create a wrapper function that takes the time stamp, access key and encryption key as arguments, and generates a signature immediately before calling the endpoint.

How often do I need to regenerate the oAuth 2.0 token?

Once generated the oAuth token is valid for 1 hour and can be used for multiple requests.

Is the API throttled?

Yes, in two ways. First, you can execute 300 requests per minute per account (but a single request can retrieve one series or a basket containing thousands of series). You will receive "HTTP 429 Too Many Requests" error. Second, you can retrieve only one gigabyte of data per month. This includes all of the metadata and HTTP headers, although these are insignificant relative to the data payload. The number of requests and series are not specifically limited.

Can I use the API to populate a data warehouse?

Yes. You may create a data warehouse for internal use, but the number of users who may have access to it is stipulated by your contract; please contact your Moody's Analytics sales representative if you have questions.

What kind of Scenario Studio information can I retrieve?

The API can return Scenario Studio project, scenario, and series details and metadata, as well as series data.

Can I create or alter a project or scenario?

Yes. The Scenario Studio v2 API now includes endpoints that enable project creation, scenario editing, and solving.

How should date parameters be constructed?

Date parameters sent to API endpoints should be integers, defined as the number of periods since Dec 31 1849. For a quarterly series, this translates to 1850Q1=1, 1850Q2=2, etc. For a monthly series this is Jan1850=1, Feb1850=2, etc.

How many series can I retrieve with the data-series endpoint?

There is no hard limit on the number of series that can be retrieved with the POST version of the data-series endpoint. However, there is a practical limit when requests become too large there is a chance of a error 504 gateway timeout. When using the GET version of the data-series endpoint, there is the additional limit of query string length.

If I alter the name of a project using the Scenario Studio web application, do I need to change my code?

No. The /project/{projectid} endpoint identifies a project by an immutable alphanumeric GUID that is assigned by our system, not the human-readable title assigned by you.

Whom do I contact for assistance in using the API?

Please go to the Economy.com Contact Us page for email, chat, and telephone options. If using the email form, set Topic to "Technical Issue."

Do other Moody's Analytics products have APIs?

Yes. We also provide APIs for our Data Buffet, AutoCycle, and Précis products.

I don't understand this jargon—can you translate?

Please see if the glossary in this document helps. It lists terminology pertaining to web APIs and related Moody's Analytics products.

Appendix 1: API endpoints

All API endpoints below are relative to the root URL https://api.economy.com/scenario-studio/v2/.

Project and scenario management end points

These endpoints allow for the creation, modification, metadata retrieval, and solving of scenarios within projects.

HTTP	Endpoint	Description
Project		

HTTP	Endpoint	Description
GET	/project	Gets the list of projects to which the user has access.
GET	/project/search	Gets the search records when searching for projects to which the user has access
GET	/project/search/count	Gets the count of search records when searching for projects to which the user has access
GET	/project/{projectId}	Gets information about a specific project.
GET	/project/{projectId}/scenario	Gets the list of scenarios within a project.
GET	/project/{projectId}/series	Gets the list of all series within a project.
GET	/project/{projectId}/geos	Gets the list of geographies within a project.
GET	/project/{projectId}/series/checked-out	Gets the list of series checked out by the current user.
GET	/project/{projectId}/series/exogenized	Gets the list of exogenized series in a project.
GET	/project/{projectId}/checkpoint/{scenarioId}	Gets the list of checkpoints for a scenario.
POST	/project/create	Creates a new project.
POST	/project/{projectId}/scenario/clone	Creates a copy of a scenario and adds it to a project.
POST	/project/{projectId}/scenario/copy	Adds a read-only copy of a scenario to a project
POST	/project/{projectId}/build	Starts the build process for a project
PUT	/project/{projectId}/settings	Update a project's settings.
PUT	/project/{projectId}/contributor/{role}	Adds contributors to a project.
DELETE	/project/{projectId}	Deletes a project.
DELETE	/project/{projectId}/scenario/alias/{alias}	Removes a scenario from a project.
Scenario		
GET	/project/{projectId}/scenario/{scenarioId}	Gets information about a specific scenario.
PUT	/project/{projectId}/scenario/{scenarioId}	Update the scenario options.
PUT	/project/{projectId}/scenario/{scenarioId}/checkpoint/{checkpointId}	Creates backup checkpoint of the scenario, and restores the given checkpoint.
POST	/project/{projectId}/scenario/{scenarioId}/solve/local	Performs a local solve on a scenario.
POST	/project/{projectId}/scenario/{scenarioId}/solve/central	Performs a central solve on a scenario.
POST	/project/{projectId}/scenario/reendogenize	Performs an add-factor solve.
POST	/project/{projectId}/scenario/{scenarioId}/checkpoint	Creates new checkpoint.

Series-related end points

These end points enable data and metadata retreival for series data. In addition, they enable editing data, equations, and state. Custom variables can also be added, modified, and deleted.

HTTP	Endpoint	Description
DataSeries		

HTTP	Endpoint	Description
POST	/project/{projectId}/data-series	Gets multiple series and/or expresssions
PUT	/project/{projectId}/scenario/{scenarioId}/data-series/{variableId}/data/local	Writes series data to a user's local copy of a scenario
PUT	/project/{projectId}/scenario/{scenarioId}/data-series/add-factor/local	Zero out add factors for a list of series in a user's local copy of a scenario
Series		
GET	/project/{projectId}/scenario/{scenarioId}/{variableId}	Gets basic information about a specific series.
GET	/project/{projectId}/scenario/{scenarioId}/variable/{variableId}	Gets all meta information for a series.
GET	/project/{projectId}/scenario/{scenarioId}/series/{variableId}/sharedown	Gets sharedown information for a series.
POST	/project/{projectId}/scenario/{scenarioId}/series/info	Gets series info for a list of series/expressions.
POST	/project/{projectId}/scenario/{scenarioId}/series/custom	Creates new custom series.
POST	/project/{projectId}/scenario/{scenarioId}/series/{variableId}/sharedown	Checks out all variables within a sharedown chain, and starts the sharedown calculation.
POST	/project/{projectId}/scenario/{scenarioId}/series/checkout	Claim out series.
POST	/project/{projectId}/scenario/{scenarioId}/series/checkin	Releases a series/unlocks a series without writing data to central.
POST	/project/{projectId}/scenario/{scenarioId}/series/commit	Push to central/commit a series-writes local data to central.
PUT	/project/{projectId}/scenario/{scenarioId}/series/exogenize	Sets series' state to exogenous.
PUT	/project/{projectId}/scenario/{scenarioId}/series/exogenize-through	Sets series to be partially exogenized.
PUT	/project/{projectId}/scenario/{scenarioId}/series/endogenizeBulk	Sets series' state to endogenous.
PUT	/project/{projectId}/scenario/{scenarioId}/series/{variableId}/equation	Edits the equation of a series.
PUT	/project/{projectId}/scenario/{scenarioId}/series/custom	Edits a custom series.
PUT	/project/{projectId}/scenario/{scenarioId}/series/{variableId}/historical/{lastHistorical}	Changes the last historical end point for a series.
DELETE	/project/{projectId}/scenario/{scenarioId}/series/custom	Deletes custom series.

Miscellaneous endpoints

HTTP	Endpoint	Description
Order		
GET	/project/{projectId}/order/{orderId}	Gets information on any single order.
GET	/project/{projectId}/order/{orderId}/build	Checks the project build order status.
SeriesSearch		

HTTP	Endpoint	Description
POST	/project/{projectId}/search/count	Gets the record count for a series search query.
POST	/project/{projectId}/search/results	Gets the records for a series search query.
Audit		
GET	/audit/project/{projectId}/count	Gets the record count for an audit search query.
GET	/audit/project/{projectId}	Gets the records for an audit search query.
Universe		
GET	/group/client	Gets the list of permissionable clients (in the context of the person making the request).
GET	/base-scenario/{scenarioId}	Gets base/Moody's scenario detail.
GET	/base-scenario	Get the list of base/moody's scenarios.
GET	/base-scenario/{scenarioId}/details	Gets the full details on a base scenario.
POST	/base-scenario/search	Gets the records for a base scenario search query.
POST	/base-scenario/search/count	Gets the record count for a base scenario search query.
POST	/project/scenario/search	Gets the records for a client scenario search query.
POST	/project/scenario/search/count	Gets the record count for a client scenario search query.

Appendix 2: HTTP server response codes

Codes returned by the Scenario Studio API are adaptations of standard HTTP server response codes.

Code	Meaning
Success codes	
200 OK	The request has succeeded
304 Not modified	The edits transmitted to Scenario Studio did not differ from teh data already on the server, thus nothing was changed
Error codes	
401 Unauthorized	The authenticating HMAC signature is outdated or the oAuth token has expired. You must generate a new signature or access token (see Authentication section).
404 Not Found	The URL path used was not found. Check the URL you are transmitting in your API request
429 Too Many Requests	You have exceeded the 300 request per minute rate limit. Throttling is access key-specific.
500 Internal Server Error	Server error - check your POST/PUT payload or query string arguments.
504 Gateway Timeout	The request is too large. Consider breaking the request into batches.

Integer codes

Frequencies

- 128 - Monthly
- 172 - Quarterly
- 204 - Annual

Variable types

- 0 - Stochastic
- 1 - Identity
- 2 - Exogenous

Variable states

Variable state is a bitwise argument in series search. Add codes together to combine.

- 1 - Endogenous
- 2 - Exogenous
- 4 - Partially exogenous

Contributor roles

- 0 - Inactive: No access to the project
- 1 - Administrator: Read/write/solve access; access to configure project, scenario, and user access settings
- 2 - Editor: Read/write/solve access
- 3 - Observer: Read-only access

Claim status (checkedOut)

Claim status is a bitwise argument in series search. Add codes together to combine.

- 1 - Unclaimed
- 2 - Claimed by others
- 4 - Claimed by you

Further reading

API documentation and functionality

- API key management
- Technical user guide
- Code samples in C#, Java, Python, R
- How to authenticate (See Authentication section)

Glossary

access key: Part of the credentials used to access the Scenario Studio API. A unique 36-character hexadecimal string, which is combined with the encryption key (qv) to produce the signature (qv).

API: Application programming interface. Generically, a set of function signatures (input and output parameters) to perform documented behavior. See also: web API (qv).

AutoCycle: See: Moody's AutoCycle™ (qv).

Coordinated Universal Time: A civil time standard based on atomic clocks and astronomical measurements, and an associated representation using a 24-hour clock that includes year, month, day, hour, minute and second, and fixed punctuation. The format is yyyymm-ddThh:mm:ssZ, for example, 2018-07-30T21:03:28Z. This format is used when making requests to the Scenario Studio API(qv). A.k.a. universal coordinated time, universal time coordinated, UTC.

cURL: Client for URLs. An open-source command-line software application to demonstrate HTTP (qv) requests and responses. Its syntax is often used to concisely document the behavior of web APIs (qv). See: curl.haxx.se

CreditCycle: see: Moody's CreditCycle™ (qv).

CSV: Comma-Separated Value. A file format that consists of plain text, where fields are separated by comma characters, and records are separated by line breaks.

encryption key: Part of the credentials used to access the Scenario Studio API. A unique 36-character hexadecimal string, which is combined with the access key (qv) to produce the signature (qv).

end point: In a web API (qv), a unique, static URL that represents an object or collection of objects; to interact with these resources, you point an HTTP client (qv) at the endpoint.

GUID: Globally Unique Identifier. GUIDs are used in enterprise software development as database keys, component identifiers, and in COM programming; they are generated by individual users with an algorithm that virtually guarantees uniqueness. A GUID is a 128-bit integer, commonly expressed as a 32-character hexadecimal string delimited by hyphens. In the Scenario Studio API, access and encryption keys, and basket and order identifiers, are GUIDs. A.k.a. Universally Unique Identifier, UUID.

HMAC: Hash-based Message Authentication Code. An international software standard (RFC2104 et seq) to verify the integrity of information transmitted over an unreliable medium such as the internet.

HTTP: HyperText Transfer Protocol. An international software standard (RFC2616 et seq) for an application-layer, client-server, stateless protocol for transmitting hypermedia documents and control information. See: https://www.w3.org/Protocols/, https://developer.mozilla.org/en-US/docs/Web/HTTP

HTTP client: Software that can communicate via HTTP (qv) with a server, for example, a web browser, cURL (qv), or a custom application that queries a web API (qv).

JSON: JavaScript Object Notation: An international software standard (ECMA-404), a lightweight data-interchange format that is easy for software to parse and generate, for humans to read and write, and is programming language-independent. JSON is the format in which the Scenario Studio API (qv) delivers individual time series (qv) and basket output (qv). See: www.json.org.

MIME: Multipurpose Internet Mail Extension. An international software standard (RFC2045 et seq) that identifies how a file transmitted over the internet (as by email

or HTTP) should be interpreted by the recipient.

metadata: Structured data that describes other data.

Moody's AutoCycle™: A software solution to forecast car prices, incorporating economic data and scenarios from Moody's Analytics. See: https://www.economy.com/products/data/autocycle

Moody's CreditCycle™: A software solution to model consumer credit risk; it combines customer data, economic data from Moody's Analytics, and consumer credit data from Equifax. See: https://www.economy.com/products/consumer-credit-analytics/moodys-creditcycle

OAuth: An open software standard (RFC5849 et seq) for services over HTTP to provide "secure delegated access" whereby server owners authorize third-party access without the clients sharing their credentials.

observation: Each numeric measurement in a time series (qv).

rate limiting: With a web API (qv), a policy that controls how many requests from a given user will be processed per unit of time, typically for billing purposes or to promote adequate performance for all users.

SHA256: Secure Hash Algorithm. A cryptographic hash function that produces a 256-bit (32-byte) output.

signature: A cryptographic string generated from the access key (qv), encryption key (qv), and a time stamp and transmitted to a web API (qv) that uses HMAC (qv) authentication. See also: SHA256 (qv).

throttling: See: rate limiting.

time series: Generically, a vector of measurements (observations [qv]) at periodic intervals. In Scenario Studio (qv), a data object that contains numeric values, metadata (qv) fields that explain how to interpret (frequency [qv] , etc.) and identify it (description, source), and one or more identifying mnemonics (qv).

UTC: See: Coordinated Universal Time (qv).

web API: A programmatic, server-side interface consisting of one or more endpoints (qv), typically expressed in JSON (qv) or XML, and exposed to the web, typically by an HTTP server.

Support

Please contact the Scenario Studio API team at Moody's Analytics by email at help@economy.com, with a subject line of "Scenario Studio API technical inquiry"

About Moody's Analytics

Moody's Analytics helps capital markets and credit risk management professionals worldwide respond to an evolving marketplace with confidence. With its team of economists, the company offers unique tools and best practices for measuring and managing risk through expertise and experience in credit analysis, economic research, and financial risk management. By offering leading-edge software and advisory services, as well as the proprietary credit research produced by Moody's Investors Service, Moody's Analytics integrates and customizes its offerings to address specific business challenges.

Concise and timely economic research by Moody's Analytics supports fi rms and policymakers in strategic planning, product and sales forecasting, credit risk and sensitivity management, and investment research. Our economic research publications provide in-depth analysis of the global economy, including the U.S. and all of its state and metropolitan areas, all European countries and their subnational areas, Asia, and the Americas. We track and forecast economic growth and cover specialized topics such as labor markets, housing, consumer spending and credit, output and income, mortgage activity, demographics, central bank behavior, and prices. We also provide real-time monitoring of macroeconomic indicators and analysis on timely topics such as monetary policy and sovereign risk. Our clients include multinational corporations, governments at all levels, central banks, financial regulators, retailers, mutual funds, financial institutions, utilities, residential and commercial real estate firms, insurance companies, and professional investors.

Moody's Analytics added the economic forecasting firm Economy.com to its portfolio in 2005. This unit is based in West Chester PA, a suburb of Philadelphia, with offi ces in London, Prague and Sydney. More information is available at www.economy.com.

Moody's Analytics is a subsidiary of Moody's Corporation (NYSE: MCO). Further information is available at www.moodysanalytics.com.

DISCLAIMER: Moody's Analytics, a unit of Moody's Corporation, provides economic analysis, credit risk data and insight, as well as risk management solutions. Research authored by Moody's Analytics does not reflect the opinions of Moody's Investors Service, the credit rating agency. To avoid confusion, please use the full company name "Moody's Analytics", when citing views from Moody's Analytics.

About Moody's Corporation

Moody's is an essential component of the global capital markets, providing credit ratings, research, tools and analysis that contribute to transparent and integrated financial markets. **Moody's Corporation** (NYSE: MCO) is the parent company of Moody's Investors Service, which provides credit ratings and research covering debt instruments and securities, and **Moody's Analytics**, which encompasses the growing array of Moody's nonratings businesses, including risk management software for financial institutions, quantitative credit analysis tools, economic research and data services, data and analytical tools for the structured finance market, and training and other professional services. The corporation, which reported revenue of $3.6 billion in 2016, employs approximately 11,500 people worldwide and maintains a presence in 41 countries.