Backend API Version 1.0 Specification

Initial Draft Version 1.0

Author: *jd*
Last reviewed: *December 26, 2022*

Table of Contents

1. Document Objectives and Purpose

This document contains the API specification for the web services provided by the *backend* that were developed, i.e., the F*orecast* and the T*rain Model* services.

2. Notation

Table 1 Notation used in this document.

Symbol	**Description (Meaning)**
R	*Required. The corresponding field must contain a value.*
List[integer|string]	*A list (array) of integer or a list of string.*

3. Restrictions

Some restrictions may apply. This is the place where restrictions should be presented and discussed.

4. Assumptions

Some assumptions may apply. This is the place where assumptions should be presented and discussed.

A1: *The backend web services assume that a request to obtain a forecast or to train a model, always correspond to one and only one task. If an execution plan has M >1 forecast or train model tasks, M requests must be performed. The backend does not guarantee the order in which requests are processed and results are returned to the client.*

5. Forecast Service API

The *Forecast Service* allows obtaining forecasts using a previously trained model. The models used are stored on the computer's hard disk.

5.1 /api/v1/models/{*model_id*}/forecast

Allows obtaining a forecast using a previously trained model identified with a model Id.
Type: **POST**.

5.1.1 Access Rules

This *endpoint* (*route*) expects an *HTTP Authorization Scheme: bearer* to be used. In the current implementation, the *endpoint* expects a *jwt bearer* with a specific value (this can change in the future).

5.1.2 Input

Path Parameters

Parameter Name	Type	Value	Description
model_id	*integer*	*R*	Identifier of the model to be used.

Request Body Schema: application/json

Parameter Name	Type	Value	Description
client_id	*string*	*R*	Identifier of the client making the request.
model_input_data	*dictionary*	*R*	Input the model needs to compute a forecast.
forecast_period	*integer*	*R*	Forecast period in hours.

Notes:
- Since two models may need different inputs to compute a forecast, the type of *model_input_data* is variable (*dictionary*).
- The forecast period can be obtained from the backend database (this parameter may be removed in the future).
- A client μ can only use a model trained by client μ.

5.1.3 Output

Response Schema: application/json

Parameter Name	Type	Value	Description
forecast	*List[integer]*	*R*	A list (array) with the values predicted. The size of the list is equal to the forecast period specified.

5.1.4 Error Messages

In case an error occurs, an *HTTP Exception* will be raised and sent to the client.

Response Schema (422 Validation Error): application/json

Parameter Name	Type	Value	Description
detail	*List[Detail]*	*R*	A list (array) of objects of type *Detail (See next table)*.

Detail

Parameter Name	Type	Value	Description
loc	*List[integer|string]*	*R*	Location of the error.
msg	*string*	*R*	Message associated with the error.
type	*string*	*R*	Type of error.

Notes:
- This validation error is raised automatically by the underlying framework in response to an invalid request input.

Response Schema (HTTP Exception): application/json

Parameter Name	Type	Value	Description
status_code	*string*	*R*	HTTP error code, e.g., 404, 400, 500.
detail	*string*	*R*	Error message, e.g., Client μ not found!

Notes:

- Various *HTTP Exceptions* are used to report errors (if this is not ideal, errors can be reported using a different mechanism/different messages).

5.1.5 Input for the Models Supported

MLPRegressor

Input Parameter	Type	Value	Description
model_input_data	*List[integer]*	*R*	List of known values (previous observations). This list must have a size equal to the *forecast_period* specified in the forecast request.

Commented [JD1]: Needs confirmation.

HistGradientBoostingRegressor

Input Parameter	Type	Value	Description
model_input_data	*dictionary*	*R*	A *dictionary* with the form: *model_input_data = {* *'Month': List[integer],* *'Day': List[integer],* *'Hour': List[integer]* *};* In the current implementation, the *List[integer] objects* in the *dictionary*, must have a size equal to the *forecast_period* specified in the forecast request.

Commented [JD2]: Can be optimized in the future.

Commented [JD3R2]: Also needs confirmation.

Important Note:

- In the case of the *HistGradientBoostingRegressor* model, the labels used in the *dictionary*, i.e., *Month*, *Day*, and *Hour*, must be presented exactly in the same way as the labels used during the training phase, or an error will be raised by the model.

5.2 Additional Information

Use the following links to get more information about (or to interact with) the API:

- http://<*ip*>:<*port*>/docs
- http://<*ip*>:<*port*>/redoc
- http://<*ip*>:<*port*>/openapi.json

where <*ip*>:<*port*> refers to the ip and the port where the service is running (these links need internet access to work).

6. Train Model Service API

The Train Model *Service* allows training different types of machine learning models. The models trained are stored on the computer's hard disk and can be used later to obtain forecasts. Trained models are associated with one and only one client. A trained model can only be used by the client it is associated with.

6.1 /api/v1/models/{*model_type*}/train

Allows training a model of the type specified in the *model_type* path parameter variable if the type specified is available. A message is sent to the client informing that the task was accepted, or an error occurred. The model is trained in a background task. That is, a model may not immediately be available or trained.

Type: **POST**.

6.1.1 Access Rules

The same as the forecast endpoint (*See Section 5.1.1*).

6.1.2 Input

Path Parameters

Parameter Name	Type	Value	Description
model_type	*string*	*R*	Type of model to train. See Section 6.1.5 for a list of the types of models available.

Request Body Schema: application/json

Parameter Name	Type	Value	Description
client_id	*string*	*R*	Identifier of the client making the request.
model_name	*string*	*R*	A name for the model, e.g., *Schedules for Store A*.
input_data	*dictionary*	*R*	Input needed to train the model (See Section 6.1.6 for more details).
forecast_period	*integer*	*R*	Forecast period in hours that will be associated with the model.

Notes:

- The *model_name* is something that makes sense to the client and is not used internally by the service to make decisions about how the model will be stored or identified.
- Since different inputs may be needed to train different models, the input data used for training is expected to be encapsulated in a *dictionary* data type (See Section 6.1.6 for more details).

6.1.3 Output

Response Schema: application/json

Parameter Name	Type	Value	Description
detail	*string*	*R*	If the task is accepted: 'detail': '1'.
task_id	*integer*	*R*	The task id.

6.1.4 Error Messages

In case an error occurs, an *HTTP Exception* will be raised and sent to the client as in the case of the Forecast Service (See Section 5.1.4).

6.1.5 Types of Models Supported (Available)

Input Parameter	Model Type	Package/Toolkit
model_type	*MLPRegressor*	sklearn.neural_network
	HistGradientBoostingRegressor	sklearn.ensemble

6.1.6 Input Data Structures for Training

Model	Input Data Structure for Training

MLPRegressor *HistGradientBoostingRegressor*	*input_data = {* *'ds': List[string],* *'values': List[integer]* *};* where *ds* is a list of *strings* representing *datestamps*, e.g., '*2019-04-13 09:15:00'*, and *values* are the observations corresponding to the *datestamps*. An implicit *1-1* correspondence between the elements of the two lists is expected, i.e., the two lists must have the same size. In general, the size of the input data for training should be significantly bigger than the *forecast_period* specified. These details, however, will not be discussed in this document.

6.2 Additional Information

See Section 5.2.

7. Additional Services API

7.1 /api/v1/models

List of supported (available) models for training.
Type: **GET**.

7.1.1 Input

No input needed.

7.1.2 Output

Response Schema: application/json

Parameter Name	Type	Value	Description

models	*List[dictionary]*	*R*	A list (array) of *dictionary* containing information about the models available. A *dictionary* has the following structure:
			
			{
			"*model_type*": *string*,
			"*model_name*": *string*
			}
			
			Output Example:
			
			"*models*": [
			{
			"*model_type*": "*MLPRegressor*",
			"*model_name*": "*MLPRegressor*"
			},
			{
			"*model_type*": "*HistGradientBoostingRegressor*",
			"*model_name*": "*HistGradientBoostingRegressor*"
			}
			]

7.2 /api/v1/models/{*model_id*}

Details of a model previously trained.
Type: **GET**.

7.2.1 Input

Path Parameters

Parameter Name	Type	Value	Description
model_id	*integer*	*R*	The unique model identifier.

7.2.2 Output

Response Schema: application/json

Parameter Name	Type	Value	Description
id	*integer*	*R*	Model unique identifier.
type	*string*	*R*	Model type.
model_name	*string*	*R*	Model name defined by the client.
time_trained	*string*	*R*	A string with the *datestamp* when the model was trained.
metrics	*string*	*R*	A *csv* string with metrics collected during the training phase.
forecast_period	*integer*	*R*	Forecast period associated with the model.
train_params	*string*	*R*	A *csv* string with custom train parameters used in the training phase.

Output Example:

"*id*": 21,
"*type*": "MLPRegressor",
"*model_name*": "Sales Store A",
"*time_trained*": "2022-12-25T17:30:18",
"*metrics*": "rmse:10.996;mae:4.960;r2_score:0.958;",
"*forecast_period*": 720,
"*train_params*": "max_number_iterations:10;"

Errors List

Error	Message
model not found	"*status_code*": 404, "*detail*": "Model {*model_id*} not found!"

7.2.3 Discussion

- Should this API also expect (consider) the *client_id*?
- The output of this API will most likely change to contain an html report instead.

7.3 /api/v1/clients

List of 'registered' (known) clients.
Type: **GET**.

7.3.1 Input

No input needed.

7.3.2 Output

Response Schema: application/json

Parameter Name	Type	Value	Description
clients	*List[dictionary]*	*R*	A list (array) of *dictionary* containing information about the known clients. A *dictionary* has the following structure: { "*client_pkey*": *integer*,

			"*id*": *string*,
			"*culture*": *string*,
			"*is_active*": boolean
			}
			
			Output Example:
			
			"*clients*":[
			{
			"*client_pkey*": 1,
			"*id*": "AB19GF",
			"*culture*": "es-ES",
			"*is_active*": true
			},
			{
			"*client_pkey*": 2,
			"*id*": "AF52HY",
			"*culture*": "en-EN",
			"*is_active*": true
			}
			]

7.4 /api/v1/clients

Creates (registers) a new client.
Type: **PUT**.

7.4.1 Input

Request Body Schema: application/json

Parameter Name	Type	Value	Description
id	*string*	*R*	Client unique identifier used when making requests.
culture	*string*	*R*	The client's culture (language).
is_active	*boolean*	*R*	Indicates if the client is active.

7.4.2 Output

Response Schema: application/json

Outcome	Output Message	Description
success	"*detail*": "*1*"	The client was created (registered).
error	"*detail*": "Client *{client.id}* already exists!"	A client with the same id already exists!

7.5 /api/v1/clients/{*client_id*}

Updates the parameters associated with client *client_id*.
Type: **POST**.

7.5.1 Input

Path Parameters

Parameter Name	Type	Value	Description
client_id	*string*	*R*	The unique client's identifier.

Request Body Schema: application/json

Parameter Name	Type	Value	Description

culture	*string*	*R*	The client's culture (language).
is_active	*boolean*	*R*	Indicates if the client is active.

7.5.2 Output

Response Schema: application/json

Outcome	Output Message	Description
success	"*detail*": "*1*"	The client parameters where updated.
error	"*status_code*": *404*, "*detail*": "Client *{client_id}* not found!"	Client with specified *client_id* not found!

7.6 /api/v1/clients/{*client_id*}

Details of the client *client_id*.
Type: **GET**.

7.6.1 Input

Path Parameters

Parameter Name	Type	Value	Description
client_id	*string*	*R*	The unique client's identifier.

7.6.2 Output

Response Schema: application/json

Outcome	Output Message	Description
success	"*id*": *string* "*culture*": *string* "*is_active*": *boolean* "*client_pkey*": *integer* **Example:** *"id": "ZO52WE",* *"culture": "us-US",* *"is_active": true,* *"client_pkey": 8*	Client details.
error	"*status_code*": *404*, "*detail*": "Client *{client_id}* not found!"	Client with specified *client_id* not found!

7.7 /api/v1/clients/{*client_id*}/models

List of models associated with the client *client_id*.
Type: **GET**.

7.7.1 Input

Path Parameters

Parameter Name	Type	Value	Description
client_id	*string*	*R*	The unique client's identifier.

7.7.2 Output

Response Schema: application/json

Parameter Name	Type	Value	Description

models	*List[dictionary]*	*R*	A list (array) of *dictionary* containing information about the *models* associated with the client. A *dictionary* has the following structure:
			
			{
			"*id*": *integer*
			"*type*": *string*
			"*model_name*": *string*
			"*time_trained*": *string*
			"*metrics*": *string*
			"*forecast_period*": *integer*
			"*train_params*": *string*
			}
			
			Output Example:
			
			"*models*": [
			{
			"*id*": 7,
			"*type*": "MLPRegressor",
			"*model_name*": "Sales Store A",
			"*time_trained*": "2022-12-21T15:18:22",
			"*metrics*": "rmse:9.75;mae:7.36;r2_score:0.96;",
			"*forecast_period*": 720,
			"*train_params*": "max_number_iterations:10;"
			},
			{
			"*id*": 20,
			"*type*": "HistGradientBoostingRegressor",
			"*model_name*": "Schedules Store B",
			"*time_trained*": "2022-12-25T00:58:28",
			"*metrics*": "rmse:11.05;mae:5.00;r2_score:0.95;",
			"*forecast_period*": 720,
			"*train_params*": ""
			}
			]

Errors List

Error	Message
client with specified client_id not found	"*status_code*": 404, "*detail*": "Client {*client_id*} not found!"

7.8 /api/v1/clients/{*client_id*}/tasks

List of training tasks requested by client *client_id*.
Type: **GET**.

7.8.1 Input

Path Parameters

Parameter Name	Type	Value	Description
client_id	*string*	*R*	The unique client's identifier.

7.8.2 Output

Response Schema: application/json

Parameter Name	Type	Value	Description

tasks	*List[dictionary]*	*R*	A list (array) of *dictionary* containing information about the training *tasks* requested by the client. A *dictionary* has the following structure:
			
			{
			"*id*": *integer*
			"time_created": *string*
			"time_started": *string*
			"time_finished": *string*
			"model_type": *string*
			"state": *string*
			}
			
			Output Example:
			
			"tasks":[
			{
			"*id*": 7,
			"*time_created*": "2022-12-21T15:17:59",
			"*time_started*": "2022-12-21T15:17:59",
			"*time_finished*": "2022-12-21T15:18:22",
			"*model_type*": "MLPRegressor",
			"*state*": "Finished"
			},
			{
			"*id*": 12,
			"*time_created*": "2022-12-23T16:39:26",
			"*time_started*": "2022-12-23T16:39:26",
			"*time_finished*": null,
			"*model_type*": "MLPRegressor",
			"*state*": "Error"
			}
			]

Errors List

Error	Message
client with specified client_id not found	"*status_code*": 404, "*detail*": "Client {*client_id*} not found!"

7.9 /api/v1/tasks/{*task_id*}

Details of training task *task_id*.
Type: **GET**.

7.9.1 Input

Path Parameters

Parameter Name	Type	Value	Description
task_id	*integer*	*R*	The task unique identifier.

7.9.2 Output

Response Schema: application/json

Outcome	Output Message		Description
success	"*id*": "*time_created*": "*time_started*":	*integer* *string* *string*	Task details.

	"*time_finished*": *string* "*model_type*": *string* "*state*": *string* ***Example***: "*id*": *7*, "*time_created*": "2022-12-21T15:17:59", "*time_started*": "2022-12-21T15:17:59", "*time_finished*": "2022-12-21T15:18:22", "*model_type*": "MLPRegressor", "*state*": "Finished"	
error	"*status_code*": *404*, "*detail*": "Task *{task_id}* not found!"	Task with specified *task_id* not found!

7.9.3 Discussion

- Should this API also expect (consider) the *client_id*?

7.10 /api/v1/tasks/{*task_id*}/state

State of training task *task_id*.
Type: **GET**.

7.10.1 Input

Path Parameters

Parameter Name	Type	Value	Description
task_id	*integer*	*R*	The task unique identifier.

7.10.2 Output

Response Schema: application/json

Outcome	Output Message	Description
success	" *state*": *string* *Example:* "*state*": "Finished"	Task state.
error	"*status_code*": *404*, "*detail*": "Task *{task_id}* not found!"	Task with specified *task_id* not found!

7.10.3 Discussion

- Should this API also expect (consider) the *client_id*?