Using REST web services

General considerations

You can run predictions on OCHEM using simple REST-like web services.

> Some OCHEM models (e.g. models aggregated using Bagging method), require significant amount of calculations.
> Thus, prediction of even a single molecule can take a minute. However, predictions are cached and will be very fast for repetitive calls.

Way 1: "Request until done"

To predict a molecule, run the following request (as example):

https://ochem.eu/modelservice/getPrediction.do?modelId=1&mol=c1ccccc1

It will predict mutagenicity of benzene (**c1ccccc1** is **MOLECULE**) using the AMES model https://ochem.eu/model/1 (**1** is **MODEL_ID**)

In general your request should have form https://ochem.eu/modelservice/getPrediction.do?modelId=MODEL_ID&mol=MOLECULE where **MODEL_ID** is the public model identifier and **MOLECULE** is the analyzed molecule.

The **MOLECULE** can be in SMILES or SD format. For both the formats, multiple molecules can be posted using $$$$ separator.
It is *much* more efficient to predict molecules in batches rather than posting separate tasks for each molecule.

The resulting JSON will look like:

```
{
  "status" : "pending",
  "taskId" : 0,
  "metaserverTaskId" : 0
}
```

The above response means that this molecule is new and it is being calculated at the moment.

Please, repeat the request at the periodic intervals (every 5-10 seconds) until the prediction result is returned in the following format:

```
{
        "taskId": "0",

  // Array of predictions (for each input molecule)
        "predictions": [{
            "moleculeID": "1002136505",

            // Array of predictions for a given molecule. Normally, contains only one
prediction.
            // Can contain multiple predictions for multi-models
            "predictions": [{
                "unit": "-log(mmol/L)",
                // Name of the predicted class for classification models. Same as
"value" for regression models.
                "predictedValueString": "2.71",
                "value": "2.71", // Prediction value (round it as you find necessary)
                "dm": "0.86", // The "distance to model" used for the accuracy
estimation
    "insideAD" : true, // is this molecule inside the model's applicability domain?
     "property": "log(IGC50-1)", // The predicted property
                "accuracy": "0.70", // The prediction accuracy (RMSE)
                "realValue": "0.0"
            }],
            "depictionID": "1000651576"
        }],
        "metaserverTaskId": "0",
        "status": "success",
        "modelDescriptionUrl": "http://ochem.eu/model/3"
}
```

Way 2: Using task IDs

Since a prediction is not instantaneous and can take several seconds to minutes, the prediction is performed asynchronously, that is in two steps:

1. Start a prediction task and get a task ID
2. Fetch your prediction task using the task ID from step (1). Keep fetching until the task is ready

The API for these two simple steps is described below.

To post a task, run the following request:

https://ochem.eu/modelservice/postModel.do?modelId=YOUR_MODEL_ID&mol=YOUR_MOLECULE

The resulting JSON will look like:

```
{
     "taskId": "1000042989", // This is the task ID you need to know
     "metaserverTaskId": "-1",
     "status": "success"
}
```

Given that the **result.modelResponse.status** is "success", the task ID used for retrieving the predictions is **result.modelResponse.taskId**

To fetch the model, use the following request:

http://ochem.eu/modelservice/fetchModel.do?taskId=YOUR_TASK_ID

If the task is still running, the resulting JSON will look like:

```
{
     "taskId": "0",
     "metaserverTaskId": "0",
     "status": "pending" // Keep requesting at periodic intervals, while the status is
"pending"
}
```

When the task is ready, the JSON will look like:

```
{
        "taskId": "0",

  // Array of predictions (for each input molecule)
        "predictions": [{
            "moleculeID": "1002136505",

            // Array of predictions for a given molecule. Normally, contains only one
prediction.
            // Can contain multiple predictions for multi-models
            "predictions": [{
                "unit": "-log(mmol/L)",
                // Name of the predicted class for classification models. Same as
"value" for regression models.
                "predictedValueString": "2.71",
                "value": "2.71", // Prediction value (round it as you find necessary)
                "dm": "0.86", // The "distance to model" used for the accuracy
estimation
                "property": "log(IGC50-1)", // The predicted property
                "accuracy": "0.70", // The prediction accuracy (RMSE)
                "realValue": "0.0"
            }],
            "depictionID": "1000651576"
        }],
        "metaserverTaskId": "0",
        "status": "success",
        "modelDescriptionUrl": "http://ochem.eu/model/3"
}
```