

## Broad Functionality: REST API Deployment

### Principle: Deployment and Prediction

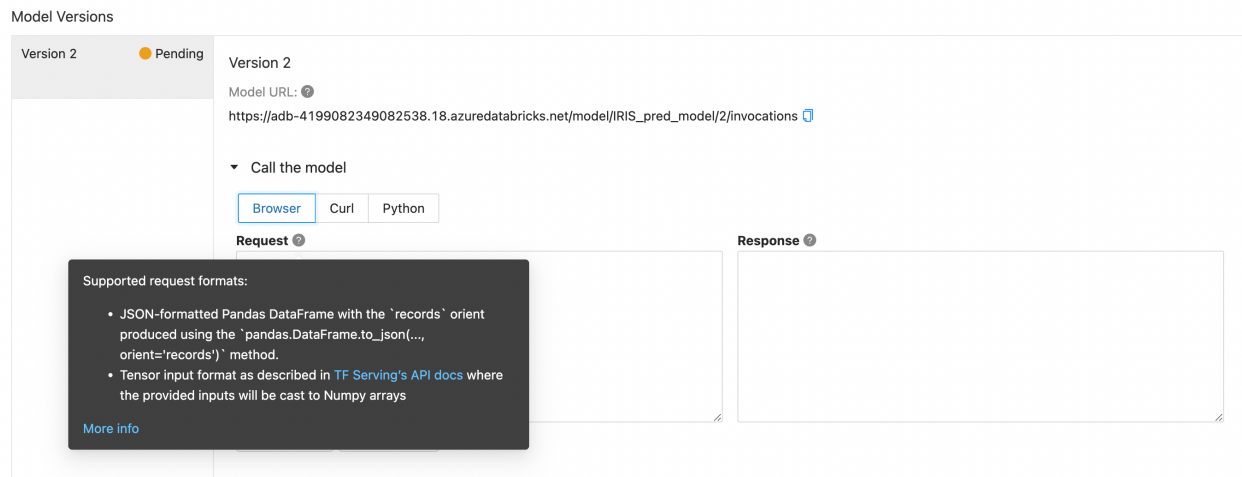
Understanding Functionality: MLflow should enable the users to deploy their models directly through the UI or the API with minimal steps and instantly get a REST API endpoint to their models, which get served through flask or other web frameworks.

### Does MLflow supports the functionality *directly*: Yes

- Explanation: The MLflow REST API allows you to create, list, and get experiments and runs, and log parameters, metrics, and artifacts. The API is hosted under the /API route on the MLflow tracking server.
- Extra remark: MLflow generates endpoints and is Available through python, bash - curl, and visualization tool 'PowerBI'. Prediction on test data can be tested on the fly, through model serving options within the Model section in data bricks.

### Does MLflow supports the functionality *indirectly*: NA

### Screenshots for MLflow supports the functionality *directly* or *indirectly*:



## Model Versions

Version 2	Pending
<p><b>Version 2</b></p> <p>Model URL: <a href="https://adb-4199082349082538.18.azuredatabricks.net/model/IRIS_pred_model/2/invocations">https://adb-4199082349082538.18.azuredatabricks.net/model/IRIS_pred_model/2/invocations</a></p> <p>▼ Call the model</p> <div> <span>Browser</span> <span>Curl</span> <span>Python</span> </div> <pre>curl \   -u token:\$DATABRICKS_TOKEN \   -X POST \   -H "Content-Type: application/json" \   -d@data.json \   https://adb-4199082349082538.18.azuredatabricks.net/model/IRIS_pred_model/2/invocations</pre>	

## Model Versions

Version 2	Pending
<p><b>Version 2</b></p> <p>Model URL: <a href="https://adb-4199082349082538.18.azuredatabricks.net/model/IRIS_pred_model/2/invocations">https://adb-4199082349082538.18.azuredatabricks.net/model/IRIS_pred_model/2/invocations</a></p> <p>▼ Call the model</p> <div> <span>Browser</span> <span>Curl</span> <span>Python</span> </div> <pre>import os import requests import numpy as np import pandas as pd  def create_tf_serving_json(data):     return {'inputs': {name: data[name].tolist() for name in data.keys() if isinstance(data, dict) else data.tolist()}}  def score_model(dataset):     url = 'https://adb-4199082349082538.18.azuredatabricks.net/model/IRIS_pred_model/2/invocations'     headers = {'Authorization': f'Bearer {os.environ.get("DATABRICKS_TOKEN")}' }     data_json = dataset.to_dict(orient='split') if isinstance(dataset, pd.DataFrame) else create_tf_serving_json(dataset)     response = requests.request(method='POST', headers=headers, url=url, json=data_json)     if response.status_code != 200:         raise Exception(f'Request failed with status {response.status_code}, {response.text}')     return response.json()</pre>	