API Contract Documentation

Frontend to Backend API Communication

Base URL: `{environment.apiUrl}` (typically `/api`, Base URL Configured with Nginx)

Authentication Controller (/api/Auth)

POST /api/Auth/login **Description:** Authenticate user credentials and return JWT token.

Request Body:

```
{
  "username": "string",
  "password": "string"
}
```

Success Response (200 OK):

```
{
  "token": "string",
  "username": "string"
}
```

Error Responses: - **401 Unauthorized:**

```
"Invalid credentials."
```

GET /api/Auth/verify **Description:** Verify JWT token and return user role information.

Headers: `Authorization: Bearer {token}`

Success Response (200 OK):

```
{
  "message": "Success",
  "role": "string"
}
```

Error Responses: - **401 Unauthorized:** Token invalid or missing

Data Controller (/api/Data)

POST /api/Data/upload **Description:** Legacy single file upload endpoint (commented out in current implementation).

Headers: `Authorization: Bearer {token}`

Request Body (multipart/form-data): - `File`: IFormFile - `UserId`: string

Success Response (200 OK):

```
{
  "datasetId": "string",
  "userId": "string",
  "parquetPath": "string",
  "totalRecords": 0,
  "numColumns": 0,
  "passRate": 0.0,
  "dateRange": {
    "start": "string",
    "end": "string"
  }
}
```

Error Responses: - **400 Bad Request:** “No file uploaded.” or “User ID is required.” - **500 Internal Server Error:** Processing failure

POST /api/Data/upload-chunk **Description:** Upload individual file chunk as part of chunked upload process.

Headers: `Authorization: Bearer {token}`

Request Body (multipart/form-data): - `file`: IFormFile (chunk data) - `uploadId`: string - `chunkIndex`: string (numeric) - `userId`: string

Success Response (200 OK):

```
{}
```

Error Responses: - **500 Internal Server Error:** Chunk upload failed

POST /api/Data/finish-upload **Description:** Finalize chunked upload by reassembling chunks and processing the complete file.

Headers: `Authorization: Bearer {token}`

Request Body:

```
{
  "uploadId": "string",
  "fileName": "string",
  "userId": "string",
  "totalChunks": 0
}
```

Success Response (200 OK):

```
{
  "datasetId": "string",
  "userId": "string",
  "parquetPath": "string",
  "totalRecords": 0,
  "numColumns": 0,
  "passRate": 0.0,
  "dateRange": {
    "start": "string",
    "end": "string"
  }
}
```

Error Responses: - 500 Internal Server Error: File processing failure

POST /api/Data/validate-ranges **Description:** Validate date ranges against stored dataset and return record counts.

Headers: `Authorization: Bearer {token}`

Request Body:

```
{
  "userId": "string",
  "datasetId": "string",
  "dateRanges": {
    "training": {
      "start": "string",
      "end": "string"
    },
    "testing": {
      "start": "string",
      "end": "string"
    },
    "simulation": {
      "start": "string",
      "end": "string"
    }
  }
}
```

Success Response (200 OK):

```
{
  "status": "Valid",
  "training": {
    "count": 0
  },
  "testing": {
    "count": 0
  },
  "simulation": {
    "count": 0
  },
  "monthlyCounts": {
    "2023-01": 0,
    "2023-02": 0
  }
}
```

Error Responses: - **400 Bad Request:** Invalid date ranges - **404 Not Found:** Dataset not found - **500 Internal Server Error:** Validation failure

Model Controller (/api/Model)

POST /api/Model/train **Description:** Train XGBoost model using specified date ranges and return performance metrics and plots.

Headers: `Authorization: Bearer {token}`

Request Body:

```
{
  "userId": "string",
  "datasetId": "string",
  "dateRanges": {
    "training": {
      "start": "string",
      "end": "string"
    },
    "testing": {
      "start": "string",
      "end": "string"
    },
    "simulation": {
      "start": "string",
      "end": "string"
    }
  }
}
```

Success Response (200 OK):

```
{
  "metrics": {
    "accuracy": 0.0,
    "precision": 0.0,
    "recall": 0.0,
    "f1Score": 0.0,
    "truePositive": 0,
    "falsePositive": 0,
    "trueNegative": 0,
    "falseNegative": 0
  },
  "plots": {
    "featureImportance": "string",
    "trainingPlot": "string"
  }
}
```

Error Responses: - 400 Bad Request: "UserId and DatasetId are required." or missing required date ranges **- 500 Internal Server Error:** Training service failure

GET /api/Model/simulation-ws (WebSocket) Description: WebSocket endpoint for real-time ML model simulation streaming. Proxies connections to FastAPI simulation service.

Protocol: WebSocket (`ws://` or `wss://`)

Connection Messages:

Start Simulation:

```
{
  "action": "start",
  "userId": "string",
  "datasetId": "string"
}
```

Stop Simulation:

```
{
  "action": "stop",
  "userId": "string",
  "datasetId": "string"
}
```

Server Messages:

Simulation Result:

```
{
  "rowIndex": 0,
  "prediction": 0,
  "confidence": 0.0,
  "actual": 0,
  "isCorrect": true,
  "timestamp": "string"
}
```

Status Messages:

```
{
  "status": "stopped"
}
```

```
{
  "status": "finished"
}
```

```
{
  "status": "stopping"
}
```

Error Messages:

```
{
  "error": "string"
}
```

Backend to ML Service Communication

Base URL: `http://localhost:8000` (Development) / `http://ml_service:8000` (Production)

CSV Processor Router (/csv)

POST /csv/upload-chunk **Description:** Receive and store individual file chunk in temporary directory.

Request Body (multipart/form-data): - `file`: UploadFile (chunk data) - `uploadId`: string (form field) - `chunkIndex`: integer (form field) - `userId`: string (form field)

Success Response (200 OK):

```
{
  "status": "chunk uploaded",
  "chunkIndex": 0
}
```

Error Responses: - 500 Internal Server Error: "Could not save chunk {chunkIndex}."

POST /csv/finish-upload **Description:** Reassemble uploaded chunks into complete file and process it to Parquet format.

Request Body:

```
{
  "uploadId": "string",
  "fileName": "string",
  "userId": "string",
  "totalChunks": 0
}
```

Success Response (200 OK):

```
{
  "datasetId": "string",
  "userId": "string",
  "parquetPath": "string",
  "totalRecords": 0,
  "numColumns": 0,
  "passRate": 0.0,
  "dateRange": {
    "start": "string",
    "end": "string"
  }
}
```

Error Responses: - 400 Bad Request: "Upload failed: Missing chunk {i}." **- 404 Not Found:** "Upload ID not found." **- 500 Internal Server Error:** "An internal error occurred during file processing."

POST /csv/validate-ranges **Description:** Validate date ranges against stored dataset and return record counts for training, testing, and simulation periods.

Request Body:

```
{
  "userId": "string",
  "datasetId": "string",
  "dateRanges": {
    "training": {
      "start": "string",
      "end": "string"
    },
    "testing": {
      "start": "string",
      "end": "string"
    },
    "simulation": {
      "start": "string",
      "end": "string"
    }
  }
}
```

Success Response (200 OK):

```
{
  "status": "Valid",
  "training": {
    "count": 0
  },
  "testing": {
    "count": 0
  },
  "simulation": {
    "count": 0
  },
  "monthlyCounts": {
    "2023-01": 0,
    "2023-02": 0
  }
}
```

Error Responses: - 400 Bad Request:

```
{
  "status": "Invalid",
  "detail": "string"
}
```

- **404 Not Found:** Dataset not found (ValueError)
- **500 Internal Server Error:** “An internal server error occurred.”

POST /csv/train **Description:** Train XGBoost model using specified date ranges and return detailed performance metrics and visualization plots.

Request Body:

```
{
  "userId": "string",
  "datasetId": "string",
  "dateRanges": {
    "training": {
      "start": "string",
      "end": "string"
    },
    "testing": {
      "start": "string",
      "end": "string"
    },
    "simulation": {
      "start": "string",
      "end": "string"
    }
  }
}
```

Success Response (200 OK):

```
{
  "metrics": {
    "accuracy": 0.0,
    "precision": 0.0,
    "recall": 0.0,
    "f1Score": 0.0,
    "trueNegative": 0,
    "falsePositive": 0,
    "falseNegative": 0,
    "truePositive": 0
  },
  "plots": {
    "featureImportance": "string",
    "trainingPlot": "string"
  }
}
```

Error Responses: - 400 Bad Request: "Request must include 'training', 'testing', and 'simulation' date ranges." or ValueError from service **- 500 Internal Server Error:** "An internal server error occurred: {error}"

WebSocket /csv/simulation-ws **Description:** Real-time streaming WebSocket endpoint for ML model simulation results.

Protocol: WebSocket (`ws://`)

Client Messages:

Start Simulation:

```
{
  "action": "start",
  "userId": "string",
  "datasetId": "string"
}
```

Stop Simulation:

```
{
  "action": "stop",
  "userId": "string",
  "datasetId": "string"
}
```

Server Messages:

Simulation Result (streaming):

```json
{
  "rowIndex": 0,
  "prediction": 0,
  "confidence": 0.0,
  "actual": 0,
  "isCorrect": true,
  "timestamp": "string"
}
```

Status Messages:

```json
{
  "status": "stopped"
}
```

```json
{
  "status": "finished"
}
```

```json
{
  "status": "stopping"
}
```

Error Messages:

```json
{
  "error": "userId and datasetId are required."
}
```

```json
{
  "error": "An unexpected error occurred during simulation: {error}"
}
```