

Backend API

API Name: **validateUserLogin**

Method: **POST**

JSON Request

- **Username** – *Required*
- **Password** – *Required*

JSON Response

- **ID** : String
- **Message** : String
- **Username** : String
- **FirstName** : String

Validations

- Validate if username and password exist
- Tables: **tbl_Account**. See Database Tables.pdf
- See JSON samples below.

Response Samples:

- *JSON Response 1:*
 - **ID**: 0
 - **Message**: Login failed. Try again.
 - **Username** :
 - **FirstName** :
- *JSON Response 2:*
 - **ID**: 1
 - **Message**: Login successful!
 - **Username** : robb07
 - **FirstName** : Robby

API Name: **createUserAccount**

Method: **POST**

JSON Request

- **Username** – *Required*
- **Password** – *Required*
- **FirstName** – *Required*

JSON Response

- **ID** : String
- **Message** : String

- [MiddleName](#) – *Optional*
- [LastName](#) – *Required*
- [DateOfBirth](#) – *Required*

Validations

- Create user account
- Validate if username already exists
- Create Patient resource in HAPI FHIR test server and save response
- Create user record in achievement table. Set current rank to default “Bronze” and current points to 0.
- Tables: **tbl_Account**, **tbl_User_Achv**. See Database Tables.pdf
- See JSON samples below.

Response Samples:

- *JSON Response 1:*
 - [ID](#): 0
 - [Message](#): Registration failed. Username already exists.
- *JSON Response 2:*
 - [ID](#): 1
 - [Message](#): Registration successful!

API Name: [createUserBGINSEntry](#)

Method: [POST](#)

JSON Request

- [EntryDate](#) – *Required*
- [MorningBG](#) – *Required*
- [AfternoonBG](#) – *Required*
- [EveningBG](#) – *Required*
- [MorningINS](#) – *Required*
- [AfternoonINS](#) – *Required*
- [EveningINS](#) - *Required*

JSON Response

- [ID](#) : String
- [Message](#) : String
- [Summary](#) : Array String

Validations

- Only 1 entry record per date. Cannot have duplicate entries per date.
- Validate if morning blood glucose, afternoon blood glucose, and evening blood glucose against the blood glucose chart reference.

- Summary array list format [“Morning BG Message1”, “Afternoon BG Message2”, “Evening BG Message3”]
- Increment user's current achievement points to 5
- Tables: `tbl_User_BG_INS`, `tbl_BG_Chart`. See Database Tables.pdf
- See JSON samples below.

Response Samples:

- *JSON Response 1:*
 - `ID`: 0
 - `Message`: Your entry has failed. Entry for 2024-10-01 date already exists.
 - `Summary` : []
- *JSON Response 2:*
 - `ID`: 1
 - `Message`: “Your entry for 2024-09-02 date has been saved!”
 - `Summary` : [“Morning Blood Glucose is on borderline level. Consult a doctor as soon as possible.”, “Afternoon Blood Glucose is on very low level. Seek medical attention as soon as possible.”, “Evening Blood Glucose is on normal level. No action needed.”]

API Name: `getUserBGINSReadings`

Method: `POST`

JSON Request

- `Username` – Required
- `FilterBy` – Required
 - `BG` or `INS`

JSON Response

- `ID` : String
- `Message` : String
- `Readings` : Array String

Validations

- Get user blood glucose (BG) or insulin readings (INS) for the last 1 <= 14 available entries
- Readings array list is sorted by ascending order of entry date
- Sample Readings array list format [{“Entry Date”, “Morning BG/INS”, “Afternoon BG/INS”, “Evening BG/INS”}, ...]
- Tables: `tbl_User_BG_INS`. See Database Tables.pdf
- See JSON samples below.

Response Samples:

- *JSON Response 1:*
 - `ID`: 0

- **Message**: No Records Found!
 - **Readings** : []
- *JSON Response 2: For Blood Glucose - mg/dol*
 - **ID**: 1
 - **Message**: ""
 - **Readings** : [{"2024-09-01","80","90","100"}, {"2024-09-02","81","91","101"}, {"2024-09-03","82","92","102"}, {"2024-09-04","83","93","103"}, {"2024-09-05","84","94","104"}, {"2024-09-06","85","95","105"}, {"2024-09-07","86","96","106"}]
- *JSON Response 2: For Insulin - units*
 - **ID**: 1
 - **Message**: ""
 - **Readings** : [{"2024-09-01","10","20","30"}, {"2024-09-02","11","21","31"}, {"2024-09-03","12","22","32"}, {"2024-09-04","13","23","33"}, {"2024-09-05","14","24","34"}, {"2024-09-06","15","25","35"}, {"2024-09-07","16","26","36"}]

API Name: **getUserAchv**

Method: **POST**

JSON Request

- **Username** – *Required*

JSON Response

- **ID** : String
- **Message** : String
- **FirstName**: String
- **CurrentRank** : String
- **CurrentPoints**: String
- **PointsToRankUp** : String

Validations

- Get user's achievements
- $\text{PointsToRankUp} = (\text{Current Rank's Max Points} + 5) - \text{Current Points}$; Except for Gold rank.
- Tables: **tbl_User_ACHV**, **tbl_ACHV_Chart**. See Database Tables.pdf
- See JSON samples below.

Response Samples:

- *JSON Response 1:*
 - **ID** : 1
 - **FirstName**: Robby

- CurrentRank : Bronze
 - CurrentPoints: 5
 - PointsToRankUp : 35
- JSON Response 2:
 - ID : 1
 - FirstName: Jane
 - CurrentRank : Silver
 - CurrentPoints: 40
 - PointsToRankUp : 35
- JSON Response 3:
 - ID : 1
 - FirstName: Mike
 - CurrentRank : Gold
 - CurrentPoints: 110
 - PointsToRankUp : Max Level!