

James Elam

SWDV 691-1

Service Re-Design: Loan Payoff Calculator

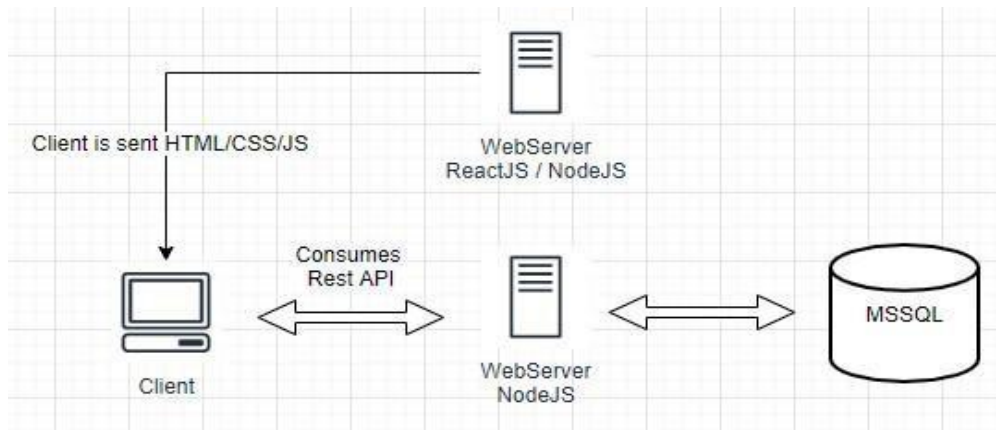
I. Preface

This service design document has been updated with changes outlined in the “Design Feedback Response” document. The “/login” endpoint has been renamed to “/sign-in” to remain consistent with “/sign-up”. I have added two new endpoints “/oauth/facebook” and “/oauth/google” which return a JWT token like “/sign-up”. Additional fields have been added to the “/sign-up” method for extend user profile attributes

As a result of these oauth endpoints, an “Endpoint Use Cases” has been added at the end of this document. The user interface had to be slightly modified on the login screen to account for signing in with Google and Facebook.

II. Technology

The Student Loan Calculator will consume a REST based API built on NodeJS with the ExpressJS framework. Users must first pass a username and password through /sign-in in order to obtain a JSON web token to authenticate on subsequent requests. Most of the POST, PUT, and DELETE methods below will directly affect the analytics and charts on the page.



III. Error Response

The Service Layer will send appropriate HTTP Status Codes depending on the error that occurred. While the application will have client-side validation, alert messages will be triggered depending on the returned Status Code.

1. Unauthorized: 401
2. Forbidden: 403
3. Resource Not Found: 404
4. Unique Constraint: 409
5. Invalid Input: 422
6. Too Many Request – 429
7. Server Error: 500

IV. Endpoints

a. POST /sign-in

i. Request Body

```
{  
  
  userName: String  
  password: String  
}
```

- ii. Returns a JSON web Token that should be used by all requests for authentication.

```
{  
  
  token: String  
}
```

b. POST /sign-up

- i. Request Body

```
{  
  
  userName: String  
  password: String  
  email: string  
  firstName: string  
  lastName: string  
}
```

- ii. Returns a JSON web Token that should be used by all requests for authentication.

```
{
```

```
token: String
```

```
}
```

c. POST /oauth/google

i. Request Body

```
{
```

```
access_token: string
```

```
}
```

ii. Response

```
{
```

```
token: String
```

```
}
```

d. POST /oauth/facebook

i. Request Body

```
{
```

```
access_token: string
```

```
}
```

ii. Response

```
{
```

```
token: String
```

```
}
```

e. GET /me

- i. There are not many major use-cases for this endpoint but can be used for user-experience type message.

- ii. Returns basic information on the currently logged in user.

```
{
```

UserID: Number

UserName: String

Email: String

LastLogin: String

DateCreated: String

```
}
```

f. GET /me/student-loans

- i. Used for displaying a list of all of the user's student loans and needed for a user to interact with them.

- ii. Returns an array of all user's student loans currently not deleted

```
[{  
  
  LoanID: Number  
  
  LoanName: String  
  
  PaymentStart: String  
  
  LoanTerm: String  
  
  StartingPrinciple: Float  
  
  CurrentPrinciple: Float  
  
  AccruedInterest: Float  
  
  InterestRate: Float  
  
  MinimumPayment: Float  
  
  StatusID: Number  
  
  IsDeleted: Boolean  
  
}, {...}, {...}]
```

d. POST /me/student-loans

- i. Create a new student-loan. This will affect the analytics and charts on the page.

ii. Post Body

```
{  
  
  LoanName: String  
  
  PaymentStart: String  
  
  LoanTerm: String  
  
  StartingPrinciple: Float  
  
  CurrentPrinciple: Float  
  
  AccruedInterest: Float  
  
  InterestRate: Float  
  
  MinimumPayment: Float  
  
  StatusID: Number  
  
}
```

iii. The newly created student-loan is returned

```
{  
  
  LoanID: Number  
  
  LoanName: String  
  
  PaymentStart: String  
  
  LoanTerm: String
```

StartingPrinciple: Float

CurrentPrinciple: Float

AccruedInterest: Float

InterestRate: Float

MinimumPayment: Float

StatusID: Number

IsDeleted: Boolean

}

f. PUT /me/student-loan/:LoanID

- i. Update an existing student loan.
- ii. The Post Body and returned entity is the same as the POST request.

g. DELETE /me/student-loans/:LoanID

- i. Deletes a single Loan
- ii. Returns a Boolean of whether or not the loan was deleted

{

LoanID: Number

Result: Boolean


```
}
```

h. GET /me/student-loans/payment-plans

i. Returns all the payment-plans created by a user.

```
{
```

PaymentPlanID: Number

PlanName: String

IsCurrent: Boolean

IsDeleted: Boolean

Payments: [

```
{
```

PaymentID: Number

PaymentDate: String

PaymentAmount: Float

AllocationMethodID: Number

IsRecurring: Boolean

IsDeleted: Boolean

```
}, {...}, {...}
```

]

}

h. POST /me/student-loans/payment-plans

i. Creates an empty payment-plan.

ii. Request Body

{

PaymentPlanID: Number

PlanName: String

}

iii. Returns the same as the GET request

h. PUT /me/student-loans/payment-plans/:PaymentPlanID

i. Updates an existing Payment Plans metadata

ii. Returns basic information on the currently logged in user.

i. DELETE /me/student-loans/payment-plans/:PaymentPlanID

i. Deletes a payment plan and all payments associated with the plan.

```
{  
  
  PaymentPlanID: Number  
  
  Result: Boolean  
  
}
```

j. POST /me/student-loans/payment-plans/:PaymentPlanID/Payment

i. Creates a new payment for an existing payment plan

ii. Request-Data

```
{  
  
  PaymentID: Number  
  
  PaymentDate: String  
  
  PaymentAmount: Float  
  
  AllocationMethodID: Number  
  
  IsRecurring: Boolean  
  
}
```

iii. Returned data

```
{
```

PaymentID: Number

PaymentDate: String

PaymentAmount: Float

AllocationMethodID: Number

IsRecurring: Boolean

IsDeleted: Boolean

}

- k. PUT /me/student-loans/payment-plans/:PaymentPlanID/Payment/:PaymentID
 - i. Deletes an existing payment associated with a PaymentPlanID

- ii. Response Data

{

PaymentPlanID: Number

Result: Boolean

}

- l. GET /me/student-loans/aggregate-analytics

- i. Returns basic statistics calculated from a user's student-loans. These are aggregate stats that change depending on the current plan selected

```
{  
  
  minimumPayOffDate: String  
  estimatedPayoffDate: String  
  currentPrincipalAmount: Float  
  totalInterestPaidCurrent: Float  
  totalInterestPaidExpected: Float  
  averageInterestAccured: Float  
}
```

- m. GET /me/student-loans/chart-analytics

- i. Returns an array of data that will power the charts.
Calculated from student-loans and current plan selected.

```
[{  
  
  Date: String  
  
  ExpectedTotalBalance: Float  
  
  Loans: [  
  
    {  
  
      LoanID: Number
```

LoanName: String

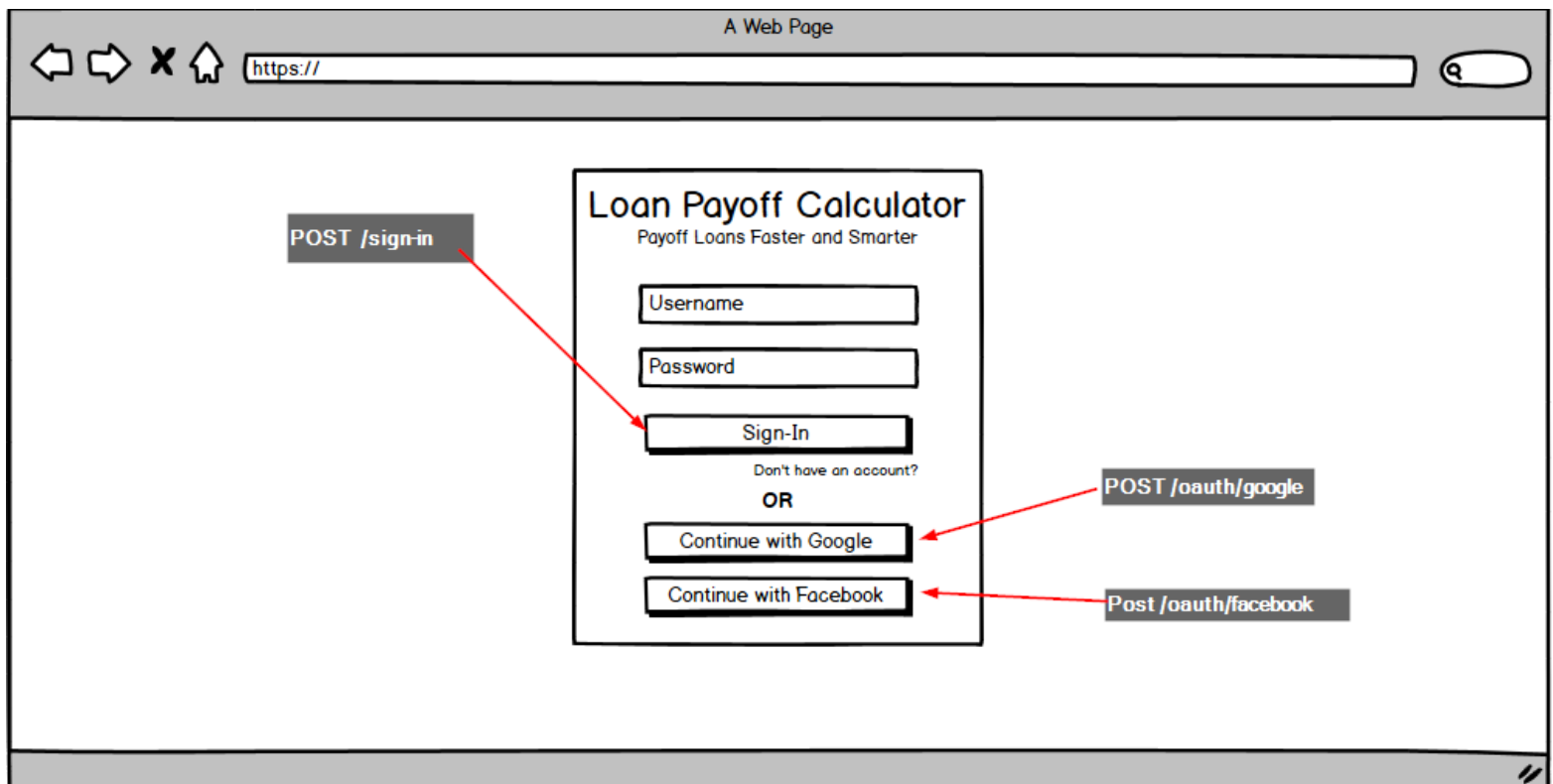
ExpectedBalance: Number

}, {...}, {...}

]

}, {...}, {...}]

IV. Endpoint Use Cases



POST /me/student-loans

Put /me/student-loans

DELETE /me/student-loans

```
graph TD; S1[POST /me/student-loans/payment-plans] --> S2[POST /me/student-loans/payment-plans/:PaymentPlanID/payment]; S1 --> S3[PATCH /me/student-loans/payment-plans/:PaymentPlanID/payment/:PaymentID]; S1 --> S4[DELETE /me/student-loans/payment-plans/:PaymentPlanID/payment/:PaymentID];
```

POST /me/student-loans/payment-plans

PATCH /me/student-loans/payment-plans/:PaymentPlanID

DELETE /me/student-loans/payment-plans/:PaymentPlanID

POST /me/student-loans/payment-plans/:PaymentPlanID/payment

PATCH /me/student-loans/payment-plans/:PaymentPlanID/payment/:PaymentID

DELETE /me/student-loans/payment-plans/:PaymentPlanID/payment/:PaymentID



Dashboard Payment Schedule Resources									
March, 2020									
Loan Name	Loan	Payment	Principle	Interest	Interest Paid	Balance	Total Principle	Total Interest	
1									
April, 2020									
Loan Name	Loan	Payment	Principle	Interest	Interest Paid	Balance	Total Principle	Total Interest	
May, 2020									
Loan Name	Loan	Payment	Principle	Interest	Interest Paid	Balance	Total Principle	Total Interest	
June, 2020									
Loan Name	Loan	Payment	Principle	Interest	Interest Paid	Balance	Total Principle	Total Interest	

GET /me/student-loans/chart-analytics

