## Holiday modifications – 18

* Configuration – 2
* Create Enum types 0 = FEDERAL, 1 = FEDERAL, 2 = BANK – 2
* Add Name, CreateDate, LastUpdate, Status (Default to 1), Type (Default to 0) to [ConnectBillPayWarehouse].[BillPay].[Holiday] table – 2
* Add Name, CreateDate, LastUpdate, Status (Default to 1), Type (Default to 0) to calendarcontroller request/response/mapping - 2
* Modify script to update existing data – SponsorId == 0 ? Federal : Bank - 2
* Dev Testing – 4
* Documentation – 1
* Peer Review - .5
* Peer Review - .5

## Configuration and Common Types - 22

* Configuration – 2
* Add createddate, lastupdateddate, category (default 0) to [ConnectBillPayCU].[dbo].[Configuration] and core request/response – 4
* Add validation to ConfigValue to Post, Put configuration endpoints based on DataType column - 2
* Add enum BILL\_PAY = 0, SYSTEM = 1, SECURITY = 2, NOTIFICATIONS = 3 – 2
* POST /api/configuration/v1/search Endpoint and service - 4
* Dev Testing – 4
* Documentation – 1
* Peer Review - .5
* Peer Review - .5

## Dashboard Metrics – 16

* Configuration – 2
* POST /api/v1/dashboard/metrics Endpoint and service – 4
* POST /api/v1/dashboard/transactions/stats– 4
* Dev Testing – 4
* Documentation – 1
* Peer Review - .5
* Peer Review - .5

## Authentication and Authorization

### Customer Management - 40

* Configuration – 2
* Create Controller and Service – 4
* GET /api/v1/customers – 4
  + Pagination?
* GET /api/v1/customers/{id} – 4
* POST /api/v1/customer – 4
* PUT /api/v1/customer/{id} – 4
* DELETE /api/v1/customer/{id} – 4
  + Verify not used in users table
* Dev Testing – 12
* Documentation – 1
* Peer Review - .5
* Peer Review - .5

### User Management - 44

* Configuration – 2
* Create Controller and Service – 4
* GET /api/v1/users – 4
  + Pagination?
* GET /api/v1/users/{id} – 4
* POST /api/v1/user – 4
* PUT /api/v1/user/{id} – 4
* DELETE /api/v1/user/{id} – 4
* GET /api/v1/users/{id}/groups - 4
* Dev Testing – 12
* Documentation – 1
* Peer Review - .5
* Peer Review - .5

### Group Management - 40

* Configuration – 2
* Create Controller and Service – 4
* GET /api/v1/ groups – 4
  + Pagination?
* GET /api/v1/groups/{id} – 4
* POST /api/v1/group – 4
* PUT /api/v1/group/{id} – 4
* DELETE /api/v1/group/{id} – 4
* Dev Testing – 12
* Documentation – 1
* Peer Review - .5
* Peer Review - .5

### Role Management - 22

* Configuration – 2
* Create Controller and Service – 4
* GET /api/v1/roles – 4
  + Pagination?
* GET /api/v1/roles/{id} – 4
* Dev Testing – 6
* Documentation – 1
* Peer Review - .5
* Peer Review - .5

### Group-Role and User-Group Assignment - 46

* Configuration – 2
* Create Controller and Service – 4
* GET /api/v1/groups/{id}/roles – 4
  + Use Group Controller and Service
* POST /api/v1/group/{id}/roles – 4
  + Use Group Controller and Service
* GET /api/v1/groups/{id}/users – 4
  + Use Group Controller and Service
* GET /api/v1/user/{id}/groups – 4
  + Use User Controller and Service
* POST /api/v1/user/{id}/groups – 4
  + Use User Controller and Service
* DELETE /api/v1/user/{id}/groups - 4
  + Use User Controller and Service
* Dev Testing – 14
* Documentation – 1
* Peer Review - .5
* Peer Review - .5

### Authentication - 52

* Configuration – 2
* Create Controller and Service – 4
* Implement [Authorize] tag method – 8
  + <https://learn.microsoft.com/en-us/aspnet/core/security/authorization/simple?view=aspnetcore-9.0>
* POST /api/v1/users/login-policy – req - 4
  + Add policy configuration in configuration table
  + Use /api/configuration to update settings in endpoint.
* POST /api/v1/users/login – req –(credentials) - 4
  + Load user permissions in cache
  + Set [authorize] permissions
* POST /api/v1/users/{token}/logout - 4
  + Clear cache and [authorize] permissions
* POST /api/v1/users/{token}/refresh – 4
* POST /api/v1/users/password – req –(token, new, old) - 4
* Dev Testing – 16
* Documentation – 1
* Peer Review - .5
* Peer Review - .5

### V2 Endpoints

Admin – 16 Endpoints (2 hours each to add authorization)

supportNotification - 4

Search - 1

Run - 1

Payee - 2

Exception - 2

CreditUnion - 4

Configuration - 1

BadRecord - 1

admin cu – 50 (2 hours each to add authorization)

Calendar - 5

Configuration - 6

Contact - 4

Exception - 2

InstitutionInfo - 3

Notfication - 7

OnUs - 4

Payee - 2

Payment - 11

PaymentHistory - 2

Report - 1

Run - 1

StatusCode – 2

Cbp.api - ? any methods called

Questions/Features

* wrgAccountNumber is not being used as far as I know in cbp. It might have been in the cbp.tools to import data from wrg
* Template activation/deactivation - Not Supported – a lot of these are used in the cbp processor and on fis incoming and cannot be disabled.
* Template versioning not supported
* Currently no support for transaction limits
* Sms not supported in cbp
* Email Delivery/Failure - Email is sent to PsiServiceHost to send. No monitoring is occurring
* Question: Definition of these - Security

- Password policy configuration

- Login policy settings

- IP whitelist management

- MFA configuration

- Audit log settings

- Security alert configuration

- Security metrics and monitoring:

- Login attempt tracking

- MFA usage statistics

- Password reset metrics

- Risk level assessments

- Security event handling:

- Risk assessment endpoints

- Security alert management

- Access attempt logging