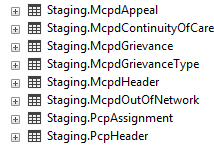
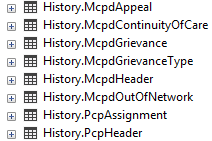
MCPDIP technical reference

1. Database designs

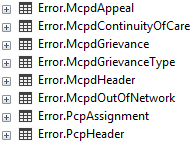
1. Staging



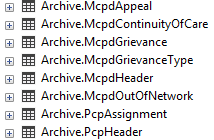
1. History



1. Error



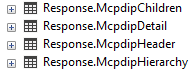
1. Archive



1. Log

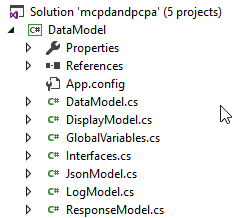


1. Response



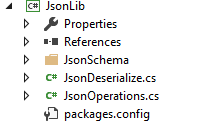
Please refer to MCPDIP\_Tables.sql, [Staging], [History], [Error], and [Archive] have same table structures, there are three log tables, Operation Log is to record important operations, like [Edit]/[Delete] records in staging tables, Process Log is more dynamic, to protect resources from multiple web users, submission log is static, will match response files and provide overall submits, accepts, and rejects, response file has the structure of headerhierarchy=>children=>detail, so the tables designed the same way

1. Solution designs
   1. DataModel



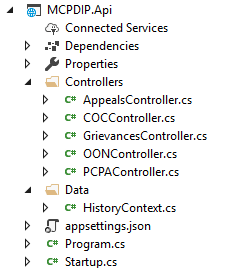
This contains all generic model classes, can be shared among multiple projects

* 1. JsonLib



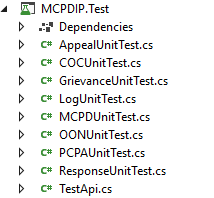
This contains all json operations, including encoding and decoding, can be shared among multiple projects

* 1. API



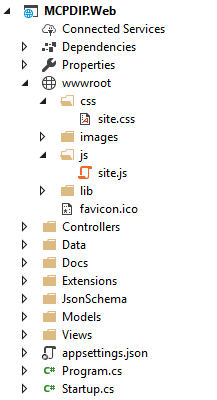
This contains RESTFUL APIs for MCPDIP data access, please refer to TestApi.cs on how to use them

* 1. Test



This contains unit testing modules come with the application, it uses in memory SQL suite to imitate dependency injections for controller

* 1. Web



Under wwwroot, only two files used, site.css and site.js

Appsettings.json is the configuration file for the entire application, it defines database connection string, json file export path, response file pickup path, etc., the settings can be changed based on the environment

Startup.cs is the module to register all bundling modules, for session, authorization and also dependency injections

Controllers is the heart of application, it has all the controllers for the page preparation, data processing and data exporting

Data folder here is just for definitions of contexts

Docs folder contains documents, including this document itself

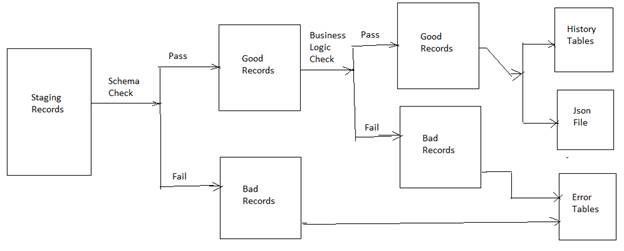
Extensions folder has HTML extensions, Session extensions and Context extensions

JsonSchema folder has all the schema files from DHCS, please make sure to update them once new schema files received from DHCS

Models folder here is for ViewModel, the carrier between views and controllers

Views folder has all the web pages

1. Process flow



Business logics will be checked after schema checks, the errors will be prefixed as

Schema Error: Business Error to differentiate errors created by HCA/HCI team

JsonController.cs has all the checks for schema errors and business errors

PCP assignment only has one business error: duplicated CIN

And for MCPD here is the list of business errors:

BL\_Grievance001: Grievance id not unique

BL\_Grievance002: Grievance id must start with 305, 306

BL\_Grievance003: Grievance receive date must be prior to current month

BL-Grievnace004: Parent grievance id logics

BL\_Appeal001: Appeal id not unique

BL\_Appeal002: Appeal id must start with 305, 306

BL\_Appeal003: Appeal receive date must be prior to current month

BL\_Appeal004: Appeal action date must be prior to current month

BL\_Appeal005: Parent appeal id logics

BL\_Appeal006: Appeal resolution date logics, must be blank if Appeal Resolution Status Indicator = Unresolved; must be populated if Appeal Resolution Status Indicator <> Unresolved; must be prior to current month; must be >== Appeal Received Date

BL\_Appeal007: Parent grievance id, if exists in history keep it, otherwise set it to null

BL\_Coc001: COC id not unique

BL\_Coc002: COC id should start with 305, 306

BL\_Coc003: COC receive date should be prior to current month

BL\_Coc004: Parent COC id logics

BL\_Coc005: Disposition indicator must not equal Provider is in MCP Network, if COC type <> MER Denial

BL\_Coc006: Expiration date must be blank if COC disposition indicator= Denied

BL\_Coc007: Expiration date must be populated if COC disposition indicator=Approved

BL\_Coc008: Denial Reason Indicator must be populated if disposition indicator = "Denied"

BL\_coc009: Denial Reason Indicator must be blank if disposition indicator<>Denied

BL\_Coc010: MER Exemption Id must be populated if COC type=MER Denial

BL\_Coc011: MER Exemption Id should be blank if COC type<>MER Denial

BL\_Coc012: exemption to enrollment denial code should be populated if COC type=MER Denial

BL\_Coc013: exemption to enrollment denial code should be blank if COC type<>MER Denial

BL\_Coc014: exemption to enrollment denial date should be populated if COC Type=MER Denial

BL\_Coc015: exemption to enrollment denial date should be blank if COC type<>MER Denial

BL\_Coc016: Exemption to enrollment denial date should be prior to current month if COC Type=MER Denial

BL\_Coc017: COC provider NPI should = submitting provider NPI, if MER COC disposition indicator<>MER COC Not Met

BL\_Coc018: MER COC Disposition indicator must be populated if COC type=MER Denial

BL\_Coc019: MEM COC disposition indicator must be blank if COC type<>MER Denial

BL\_Coc020: MER COC disposition date should be populated if COC type=MER Denial

BL\_Coc021: MER COC disposition date should be blank if COC type<>MER Denial

BL\_Coc022: MER COC disposition date should be prior to current month if COC type = MER Denial

BL\_Coc023: reason MER COC Noet Met must be populated if COC type=MER Denial and MER COC disposition indicator=MER COC Not Met

BL\_Coc024: reason MER COC Noet Met must be blank if COC type <>MER Denial

BL\_Coc025: reason MER COC Not Met must be blank if MER COC disposition indicator<>MER COC Not Met

BL\_Oon001: OON id not unique

BL\_Oon002: OON id should start with 305, 306

BL\_Oon003: OON request received date must be prior to current month

BL\_Oon004: parent OON id logics

BL\_Oon005: Partial Approval Explanation must be populated when OON Resolution Status = Partial Approval

BL\_Oon006: OON Request Resolved Date must be blank if OON Resolution Status indicator = Pending

BL\_Oon007: OON Request Resolved Date must be populated if OON Resolution Status indicator <> Pending

BL\_Oon008: OON Request Resolved Date is not a past date

BL\_Oon009: OON Request Resolved Date must be greater than or equal to OON Request Rereceived Date

1. Unit testing

Unit testing should be done within visual studio

AppealUnitTestto test Appeals controller

COCUnitTest=>to test Continuity of care controller

GrievanceUnitTest=>to test Grievance controller

LogUnitTest=>to test log controller

MCPDUnitTest=>to test MCPD json modules controller

OONUnitTest=>to test Out of Network controller

PCPAUnitTest=>to test PCP assignment controller

ResponseUnitTest=>To test response controller

TestApi=>to test restful APIs provided by the application, make sure deploy API first, please refer to the #6, deploy section on how to

1. Integration testing

Please use <http://edidev01/EJGT> for integration test

* 1. Data loading

Data loading happens in staging tables

Before loading data to staging.PcpAssignment, please update Staging.PcpHeader, like

update Staging.PcpHeader set SubmissionDate='20200810', ReportingPeriod='20200731'

Do the same thing for Staging.McpdHeader, before loading Grievance, Appeal, Continuity of Care and Out of Network data, like

update Staging.McpdHeader set SubmissionDate='20200810', ReportingPeriod='20200731'

* 1. Data viewing

Click [PCP Assignment], to view PCP assignment data in staging, history and also error tables, use [Refresh] to test searches, use [Download] to test downloads

Click [Grievance] to view grievance data in staging, history and also error tables, search modules here is a cascade dropdown to provide maximum flexibility to find every single data element

Click [Appeal] to view Appeal data in staging, history and also error tables, search modules here is a cascade dropdown to provide maximum flexibility to find every single data element

Click [COC] to view Continuity Of Care data in staging, history and also error tables, search modules here is a cascade dropdown to provide maximum flexibility to find every single data element

Click [OON] to view Out Of Nerwork data in staging, history and also error tables, search modules here is a cascade dropdown to provide maximum flexibility to find every single data element

Header data can be viewed in [PCPA Json] and [MCPD JSON]

* 1. Data editing

Editing only happens in staging tabs, for each record, there will be two links at the end, [Edit] and [Delete], please use these operations in caution, and all [Edit] / [Delete] operations will be recorded

* 1. Data Exporting

For PCP assignment, there will be json files in test mode and production mode, test file will have the same naming convention as production file, IEHP\_PCPA\_20200721\_00001.json

For MCPD, there will be json files in test mode and production mode, test file will have the same naming convention as production file, IEHP\_MCPD\_20200721\_00001.json

Inside PCP Assignment, Grievance, Appeal, COC, OON, there’s also a download module to provide instantly custom filtered raw data download, the data can be downloaded as .csv or .json

* 1. Data verification

For generated json files,

Click [PCPA File Validation], to validate the whole files

Click [MCPD File Validation], to validate the whole files

1. Deploy
   1. Critical patches

Please refer to PublishInstructions.txt, there are two critical patches must be installed before setup the web application, one is .NET core hosting bundle, and the other is .NET core SDK

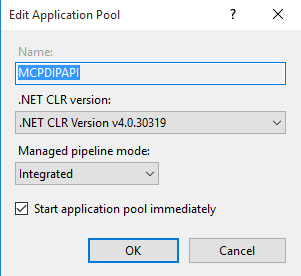
* 1. Copy the web content

Please copy the web content from [\\edidev01\ejgt](file:///\\edidev01\ejgt) to the destination server, InetPub\wwwroot\EJGT, if the path does not existed, please create the destination path

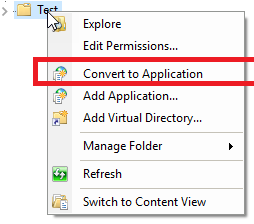
* 1. IIS settings

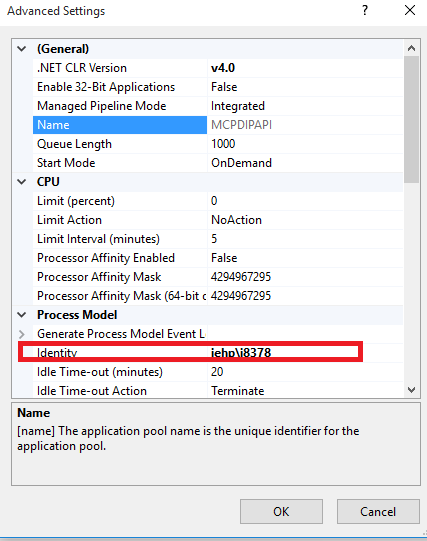
Use IIS manager to setup the web application,

First, create application pool, please set the identity to iehp\EJGTUser, password upon request



Second, convert to web application





Third, set authentications, make sure only enable Windows Authentication

