# 60i – Project Online and Project Server 2013 (EPM) Integration Architecture

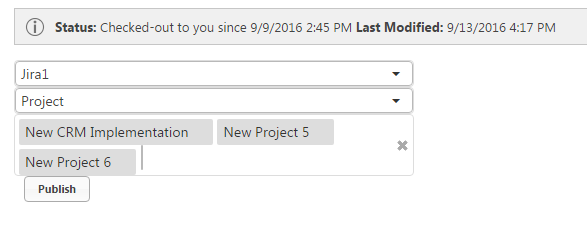
The 60i System consists of the following four parts:

**1. SharePoint add-in**

The SharePoint add-in UI Part allows connecting EPM with JIRA and VSTS.

This part is rendered as a set of combo and tag boxes and allows choosing a source Jira environment, type of connected objects (Projects or Epics) and selection of one or several objects to link. (**See Pic 1)**.

**Pic. 1**



All necessary configuration data should be stored in SharePoint list called SyncSettings. (SharePoint Layer/Content Database at the DB layer)

URL, login and password for Jira should be stored in database.

After choosing system, Epics or Projects will be loaded and will be possible to choose them in last tag box.

After pressing Publish button these values will be stored in special custom fields of current project.

**2. Database Layer**

Tables:

SyncSystemType – available system types (Jira, VSTS, etc.)

SyncSystem – list of available systems

ProjectServerSystemLink – table contains links between EPM and Jira/VSTS projects and epics

SyncSystemSetting – additional attributes for SyncSystem

SyncSystemSettingValue – values for these additional attributes

SyncSystemFieldMapping – custom fields that should be synchronized between Jira/VSTS and EPM

For example:

SystemId|SystemFieldName|EpmFieldName|FieldType|StagingFieldName|IsMultiSelect|IsIdWithValue

1 |SprintId |SixtyI\_SprintId |string |NULL |0 |0

Master, MasterFieldMappingValue – master table and custom field values. This table cleared periodically from duplicates and contains information about all records that were cached by system

Staging, StagingFieldMappingValue – staging table and custom field values. There are temporary stored records in this table. After processing system deleting them.

MasterHistory, MasterHistoryFieldMappingValue – history table. Contains all changes for all working period.

MasterWorklog – table with worklogs (for timetracking)

**3. Proxy server and Web hook receiver**.

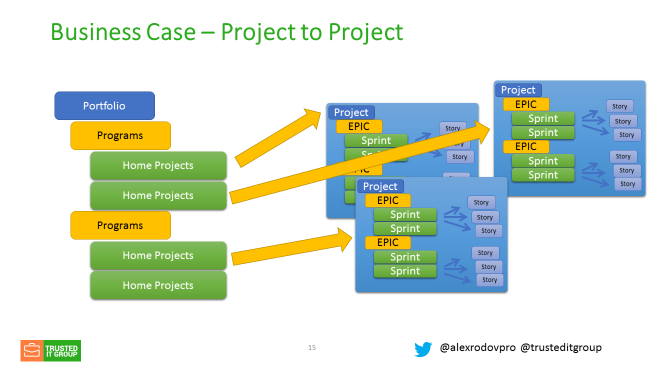
This object is running on Web Servers. Since the Jira does not allow cross-origin access, there is need to use such proxy as it enables CORS. In addition, this proxy is used to receive web hooks from Jira and VSTS.

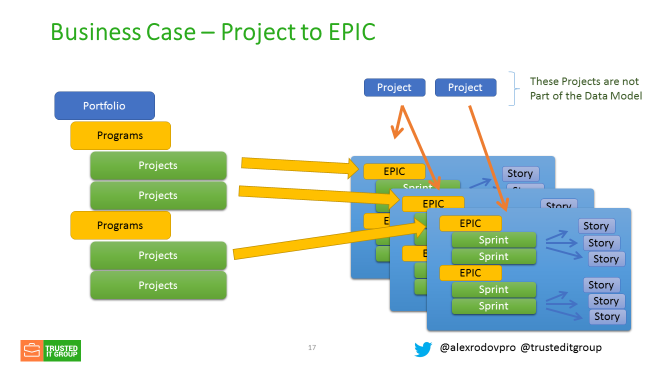
Proxy server has standard MVC architecture with Data and Business layers.

4. **Data Processors**

The data processorsrun on the application server as a windows services. They work in the background and each of them has inner timer. On timer tick, they grab unprocessed data from staging table. During processing, data from Jira is synchronized with data in EPM. Each processor does his own function:

1. ProjectOnlineSystemConnector.WinService – checkout, update (add new tasks, assignments, update tasks), publish, checkin projects.
2. If necessary, additional win services could be created.





60i modules

1. ProjectOnlineSystemConnector.HtmlCssJsReady – contains HTML, CSS and JS for 60i settings and linking pages
2. ProjectOnlineSystemConnector.Web – web site project that has 2 main functions. The first one is providing requests from 60i pages to Jira (Jira is not allowed cross-origin requests, JiraController is designed to Enable CORS and push such requests to Jira via HttpWebRequest). The second one is the receiving web hooks from Jira and write them into DB.
3. ProjectOnlineSystemConnector.Data– contains classes-entities. This classes are generated by EntityFramework.
4. ProjectOnlineSystemConnector.BusinessServices – almost all logic for communicating with DB and some helper methods for Jira.
5. ProjectOnlineSystemConnector.Common – some constants, email sender and logging methods.
6. ProjectOnlineSystemConnector.DataAccess.Database – repositories for accessing DB
7. ProjectOnlineSystemConnector.DataAccess.Jira – methods for accessing Jira
8. ProjectOnlineSystemConnector.DataAccess.CSOM – methods for accessing Project server via CSOM and OData
9. ProjectOnlineSystemConnector.Database – DB schema scripts
10. ProjectOnlineSystemConnector.DataModel – classes that is using while application is working. There are data transfer objects, classes for filling with data from Project Server (incoming OData-xml), Jira and VSTS (incoming json), some view models.
11. ProjectOnlineSystemConnector.SignalR – self hosted SignalR instance that is using to send diagnostic information from win services to 60i pages.
12. WebHookDataActualsProcessorWinService – win service that filling actuals with data from DB (without project check out-check in process)
13. ProjectOnlineSystemConnector.WinService – win service that check out, updates, publishes and check in projects.
14. ProjectOnlineSystemConnector.SyncServices – library contains main rules how data should be transferred from Jira/Tfs into DB and from DB to EPM.

