# Server Side Exercises (2 days)

# Exercise 1 - hello world

### References

ServerSideExercises.sln

HelloWorldSXPMessage.cs

# Exercise 2 – creating custom SXP processor

### References

ServerSideExercises.sln

CustomEngineerCreate.cs

### Scenario

Create a custom SXP message processor that creates a new engineer and commits it to the Database, Using the server-side business API.

# Exercise 3 – custom SXP processor

### References

ServerSideExercises.sln

CustomEngineerUpdate.cs

### Scenario

Create a custom SXP message processor that creates a new engineer and commits it to the Database, Using the server-side business API.

Create a custom SXP processor to retrieve an Engineer using SXP object reference (ID or Name properties) and update the EngineerType of the Engineer based on the value passed in the SXP message. The response should return the updated Engineer object.

# Exercise 4 – Custom SXP processor

### References

ServerSideExercises.sln

GetTasksFromDevClass.cs

### Scenario

Create a custom SXP message processor that receives a message with a name of a Region and District. The Processor will return all the tasks that their Region and District values match the values in the incoming message.

# Exercise 5 - Server API

### References

ServerSideExercises.sln

ServerEventUpdateTask.cs

### Scenario

On Task add, set the required skills & duration based on the task type

# Exercise *6 -* Server API

### References

ServerSideExercises.sln

GenericEventUpdateLateStart.cs

### Scenario

LateStart Updates:

When a Task is created check its DueDate if set then derive the LateStart time by subtracting the Duration from DueDate  
When Task DueDate is modified then derive LateStart by subtracting the Duration from DueDate

# Exercise 7

### References

CustomServiceLab.sln

CustomService.cs

### Scenario

Custom Service Lab, Implement GetResourceEx() in W6CustomService

# Exercise 8

### References

ExposeSxpAsSeviceLab.sln

CustomExposeSxpService.cs

### Scenario

‘CustomExposeSxpServiceLab’ solution, Add a new web operation called GetResourceSchedule