**Analysis of ClickCallByOperation() Method**

Contents

[Functionality Explained: 2](#_Toc476817237)

[Code Changes: 3](#_Toc476817238)

[Impact: 4](#_Toc476817239)

[Analysis Data: 4](#_Toc476817240)

[Revision History 5](#_Toc476817241)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prepared by/Modified by | | Role | | Date of preparation | |
| Syed Farhan Husain | | Consultant | | 03/09/2017 | |
| Reviewed by | | Role | | Date of Review | |
| Hita SOni | | Team Lead | | 03/09/2017 | |
| Approved by | | Role | | Date of Approval | |
|  | |  | |  | |
| Circulation List |  | | **Version number of the template** | | **0.1D** | |
| Version number of the work product | 0.1D | | | | | |

## Functionality Explained:

* **ClickCallByOperation** Method is called when a user hits request Appointment Button on the index page.
* The parameters passed to the methods are
  + earlyStart
  + lateStart
  + duration
  + profile
  + preferredFSEs
  + FSESkillLevel
  + IsExtendedSLots
  + ExtendedSlotsType
* It calculates DateTime using CountriesTimeZoneConversion class.
* It returns number of Slots for given ES and LS.
* It uses ScheduleService.ExtendedGetAppointmentsEx2Async method to retrieve slots which returns System.Threading.Tasks.Task<NewSDTApplication.ScheduleServiceDev1.ExtendedGetAppointmentsEx2Response> result which is added in collection of Tasks

While debugging the method we found that **ClickCallByOperation** is beingcalled twice for loading the appointment slots, so we went one step back and checked from which method it is being called.

It was found that ClickCallByOperation method was called from GetAppointments() method from RequestAppointmentBooking Controller.

We analyzed that why it was required to call the method twice in the same method and could one call be avoided and can we have same result after ignoring one call.

We found that each call was necessary as it was based on different scenario which is being explained below.

Here the main function of this subroutine is to retrieve slots based on ES and LS values.

Firstly we use two sets of ES and LS values to retrieve Slots.

1st set is retrieved from Click Call , we give the task duration(from SDT UI field of Task Duration) to the method ClickCallByOperation and get the ES and LS

2nd set is retrieved from the session Session["IndextoRequest"], this session is populated from the Siebel data.

The two IF conditions at the start of the method are important.

1st IF condition retrieves the slots which are “IN SLA” and the 2nd IF condition retrieves the slots which are “After SLA” and “Before SLA” with ES and LS dates.

When the slots are retrieved in the SDT UI there are 3 types of Slots namely:

1. In SLA
2. Before SLA
3. After SLA

So for all these 3 types of Slots the GetAppointment method is used.

## Code Changes:

After analyzing the method we found that there was not much scope for any change, but we found that objCountriesTimeZoneConversion = new CountriesTimeZoneConversion(); was getting initialized many times and call to objCountriesTimeZoneConversion.GetTimeZone(res.serviceRequest.countryCode, res.serviceRequest.shipToAddress); was being made on several occasions with same parameter which returned same result.

So we removed multiple initialization and at the start lines of method we took a variable dateTimeZone of DateTime Type and assigned GetTimeZoneConversion.

var res = new SiebelJsonToEntity();

if (HttpContext.Current.Session["SiebelData"] != null)

{

res = (SiebelJsonToEntity)HttpContext.Current.Session["SiebelData"];

objCountriesTimeZoneConversion = new CountriesTimeZoneConversion();

dateTimeZone = objCountriesTimeZoneConversion.GetTimeZone(res.serviceRequest.countryCode, res.serviceRequest.shipToAddress);

}

And where ever that call was made to objCountriesTimeZoneConversion.GetTimeZone(res.serviceRequest.countryCode, res.serviceRequest.shipToAddress); we replaced it with dateTimeZone variable.

## Impact:

Earlier the time it took to call CallClickByOperation from GetApointments() metho was in range on 17 to 20 seconds, after the changes it took around 14 to 16 seconds on average.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task1** | **Before( secs)** | **After ( secs)** | **Environment** |
| Click calls method optimization (ClickCallByOperation()) | 17-20 secs | 14-16 secs | Local |

## Analysis Data:

Data used for analysis is given below.

Environment: **CRP**

SR Number: **1-4763611924**

Activity ID: **1-26S4QHM**

ShipToSite ID: **1998842**

SystemID: **5973XR0010**



## Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | Date | Version # | Section / Page # changed | Details of changes made |
| 1. | 03/09/2017 | 0.1D | All | All |
| 2. |  |  |  |  |