**ReachFive**

**CIAM** integration for Salesforce Commerce Cloud Storefronts

***Version 20.4.3***





**Table of Contents**

[1. Summary 4](#__RefHeading___Toc1034_2179249197)

[2. Component Overview 5](#__RefHeading___Toc1036_2179249197)

[2.1 Functional Overview 5](#__RefHeading___Toc1038_2179249197)

[2.2 Use Cases 9](#__RefHeading___Toc1040_2179249197)

[SSO: 9](#__RefHeading___Toc1042_2179249197)

[Social Login: 9](#__RefHeading___Toc1044_2179249197)

[ReachFive Profiles Synchronization: 9](#__RefHeading___Toc1046_2179249197)

[Transition Mode: 10](#__RefHeading___Toc1048_2179249197)

[2.3 Limitations, Constraints 10](#__RefHeading___Toc1050_2179249197)

[2.4 Compatibility 10](#__RefHeading___Toc1052_2179249197)

[2.5 Privacy 10](#__RefHeading___Toc1054_2179249197)

[2.6 Security 10](#__RefHeading___Toc1056_2179249197)

[Step 1: User authentication process 11](#__RefHeading___Toc1058_2179249197)

[Step 2: Handle the Authorization Response 11](#__RefHeading___Toc1060_2179249197)

[Step 3: Exchange authorization code for ID token 11](#__RefHeading___Toc1062_2179249197)

[Step 4: Retrieve user’s profile data from ID Token 12](#__RefHeading___Toc1064_2179249197)

[3. Implementation Guide 13](#__RefHeading___Toc1066_2179249197)

[3.1 Setup 13](#__RefHeading___Toc1068_2179249197)

[3.1.1 Import Cartridge 13](#__RefHeading___Toc1070_2179249197)

[3.1.2 Import Site Preferences and Custom Objects 13](#__RefHeading___Toc1072_2179249197)

[3.2 Configuration 13](#__RefHeading___Toc1074_2179249197)

[3.2.1 Add the Reach Five Cartridge to your Storefront Cartridge Path 13](#__RefHeading___Toc1076_2179249197)

[3.2.2 Configure Reach Five Site Preferences 13](#__RefHeading___Toc1078_2179249197)

[3.2.3 Configure Reach Five Services 15](#__RefHeading___Toc1080_2179249197)

[3.2.4 Configure Reach Five Jobs 17](#__RefHeading___Toc1082_2179249197)

[Set Scope: 17](#__RefHeading___Toc1084_2179249197)

[Job Schedule (Job is scheduled to run every 5 minutes by default): 18](#__RefHeading___Toc1086_2179249197)

[3.3 Custom Code 19](#__RefHeading___Toc1088_2179249197)

[3.3.1 Custom Code. Controllers. 19](#__RefHeading___Toc1090_2179249197)

[3.3.2 Custom Code. Templates. 23](#__RefHeading___Toc1092_2179249197)

[3.4 External Interfaces 26](#__RefHeading___Toc1094_2179249197)

[Authentication 26](#__RefHeading___Toc1096_2179249197)

[Get an access token 26](#__RefHeading___Toc1098_2179249197)

[Update User 26](#__RefHeading___Toc1100_2179249197)

[3.5 Firewall Requirements 27](#__RefHeading___Toc1102_2179249197)

[4. Data Storage 28](#__RefHeading___Toc1104_2179249197)

[4.1 Availability 28](#__RefHeading___Toc1106_2179249197)

[4.2 Support 28](#__RefHeading___Toc1108_2179249197)

[5. User Guide 28](#__RefHeading___Toc3346_2179249197)

[5.1 Roles, Responsibilities 28](#__RefHeading___Toc3348_2179249197)

[5.2 Storefront Functionality 28](#__RefHeading___Toc3350_2179249197)

[Case SSO Screen Shot 29](#__RefHeading___Toc3352_2179249197)

[Register 29](#__RefHeading___Toc3354_2179249197)

[Case Social Login 29](#__RefHeading___Toc3356_2179249197)

[Login: 29](#__RefHeading___Toc3358_2179249197)

[6. Known Issues 30](#__RefHeading___Toc1110_2179249197)

[7. Failover and recovery process 30](#__RefHeading___Toc1112_2179249197)

[8. Release History 30](#__RefHeading___Toc1114_2179249197)

# 1. Summary

ReachFive is a CIAM (Customer Idendity and Access Management), providing social media and full authentication features.

With ReachFive, customers can simplify their authentication process and improve their customer identity management.

Currently, more than 30 social networks authentications are managed by ReachFive but the number of social networks supported is continuously growing.

As described in the ReachFive website (https://www.reach5.co/), ReachFive provide the following functionalities:

● Single Sign On: sets of API and widget to allow the user to signup, login, manage social account profiles

● Social Login: allows merchants to integrate OAuth providers registering and login functionalities, manage social account profiles

This component integrates SFCC with the ReachFive platform.

Using this cartridge, merchants and solution partners do not need to do anything to integrate new providers when they become available in the ReachFive platform (except activating these new providers in the ReachFive back office, as everything related to the connection is managed by ReachFive).

Merchants willing to use this component to connect to the ReachFive platform will be required to subscribe to the ReachFive service.

This component requires the generic **int\_reachfive** and **int\_reachfive\_sg** cartridges to be included in the code source of the merchant Salesforce Commerce Cloud sites. The merchants will also have to set specific Site Preferences in order for the service to work properly, as described in this documentation.

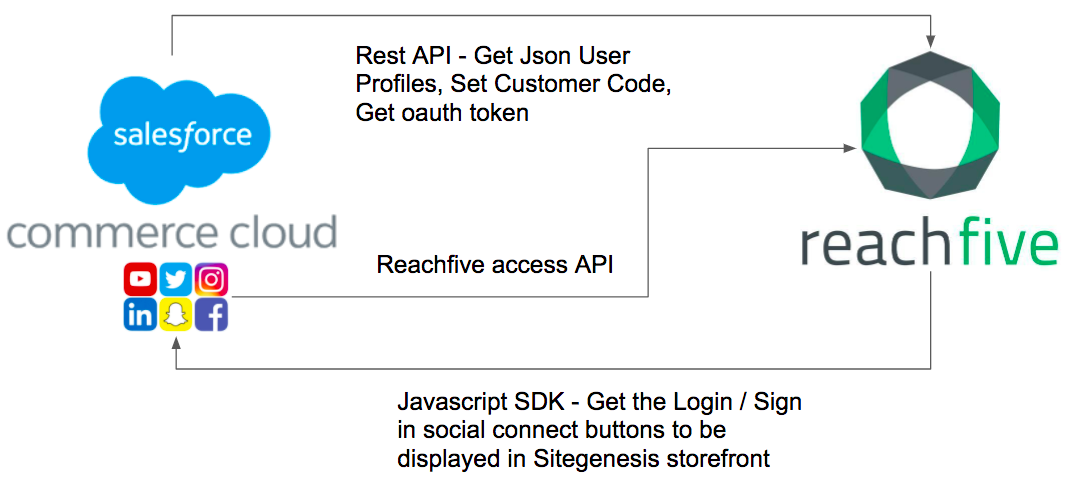
The merchant’s storefronts will be slightly modified, to include the Reach Five SSO widget or social connect buttons, as described in this documentation.

Following the current LINK certification process guidelines, the int\_reachfive\_sg cartridge has been developed in the Javascript controller mode. The development of this cartridge is based on 18.3 SiteGenesis version.

# 2. Component Overview

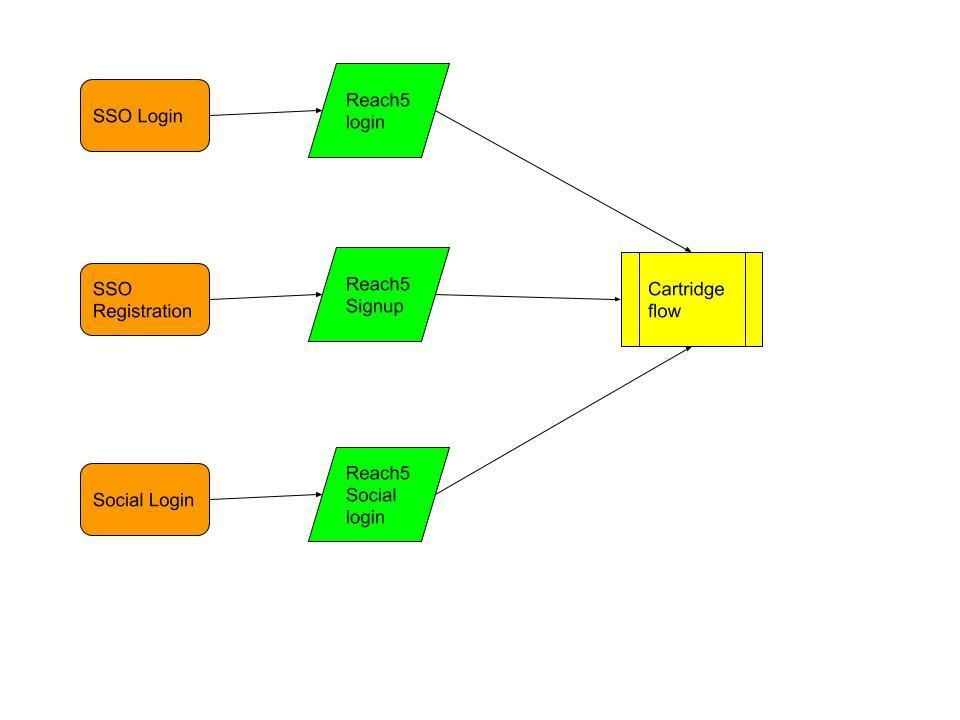
## 2.1 Functional Overview

The **int\_reachfive** and **int\_reachfive\_sg** cartridges make use of the Javascript SDK and the REST API provided by ReachFive to exchange data between the Salesforce Commerce Cloud instance and the ReachFive platform.

The Javascript SDK provides widget and API to integrate SFCC with ReachFive, in details:

* SSO: widget to manage login, signup, change password are included in the Sitegenesis login page, registration page, my account pages, checkout pages.
* Social Login: buttons for social login are added to the login page, registration page, my account page, checkout pages.

The following diagrams illustrate the Login and Registration Process

In the SSO scenario, a customer access to the SSO Login / Signup widget from:

* Login Page
* Registration Page
* Checkout Page

In these pages, a widget is presented instead of the standard sitegenesis related form.

In the SSO scenario and Social Login scenario the button for Social Login are present in the following pages:

* Login Page
* Registration Page
* Checkout Page

The SSO signup widget create a Profile in ReachFive server.

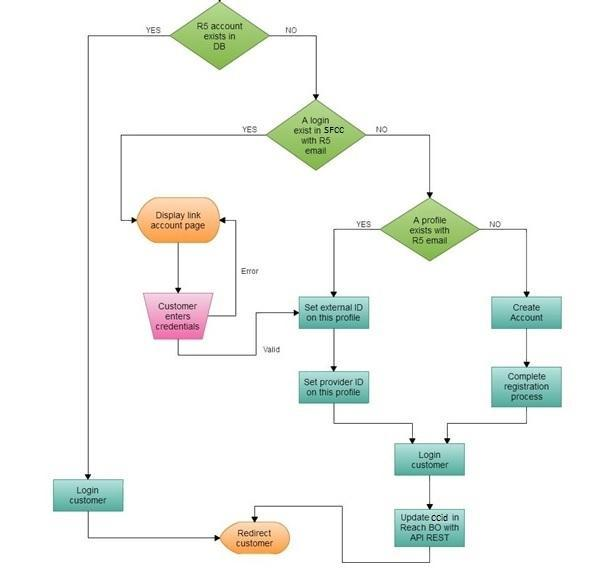
The SSO login widget search for a Profile present on ReachFive server.

The Social Login widget perform an oauth authentication with the social network account and pass the profile information to ReachFive server.

The cartridge flow process manages the link of ReachFive profile information with a customer in SFCC.

The business logic is described in the following diagram:

R5 = ReachFive

Details about ReachFive UI and Core SDK:

<https://www.npmjs.com/package/@reachfive/identity-ui>

https://developer.reachfive.com/sdk-ui/index.html

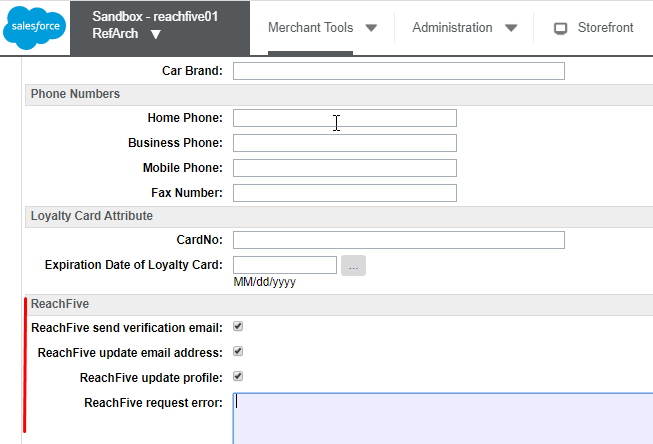
<https://www.npmjs.com/package/@reachfive/identity-core>

https://developer.reachfive.com/sdk-core/index.html

A transition phase is available in the cartridge since version 20.4.0 and allow to silently signup in Reachfive on login. It makes it possible to unplug the reachfive cartridge and keep the SFCC accounts working with the login and last password of the shoppers (password synchronized with reachfive and internal SFCC Profil).

ReachFive Profiles Synchronization

‘ReachFive-Synchronization’ job was created. The job will take all customers that have one of the Profile attributes (reachfiveSendVerificationEmail, reachfiveUpdateEmailAddress, reachfiveUpdateProfile) set to true. Then for each of those customers it will do the appropiate ReachFive calls and set the flags to false. If any of the calls fail the error will be logged in the error logs and the ‘reachfiveError’ attribute will be populated only with the last errors.

When synching the profile the job uses the “reach5ProfileFieldsJSON” Site preference in ‘ReachFive’ group to determine which fields need to be synched and mapping between SFCC and ReachFive profile fields.

The job uses such services:

* reachfive.rest.auth – to get management access token which is used in other reachFive API calls
* reachfive.updateprofile.put – to update email and profile fields
* reachfive.verifyemail.post – to send the verification email for updated email address

This job is scheduled to run every 5 minutes by default.

All issues and exceptions which occur during job running are added to job log.

## 2.2 Use Cases

The following use cases are managed by the int\_reachfive and int\_reachfive\_sg cartridges :

### SSO:

1. Registration of a new customer through a ReachFive Signup widget
2. Login of an existing SFCC Customer through ReachFive. The SFCC Customer account has been created through the native SiteGenesis account creation form.
3. Login of an existing ReachFive customer. The SFCC customer is created and linked to the ReachFive Profile
4. Social Account Activation/Deactivation, in My Account the customer using a widget can connect more than one social account to his profile.

### Social Login:

1. Registration of a new customer through a ReachFive supported social connect provider
2. Login of a customer who has previously created his SFCC Customer account through a ReachFive supported social connect provider
3. Login of an existing SFCC Customer through ReachFive. The SFCC Customer account has been created through the native SiteGenesis account creation form.
4. Login of an existing SFCC Customer through ReachFive. The SFCC Customer account has been created through a non ReachFive social connect provider (i.e through the social connect provider natively proposed by the SiteGenesis)
5. Social Account Activation/Deactivation, in My Account the customer using a widget can connect more than one social account to his profile.

### ReachFive Profiles Synchronization:

1. Update the email address
2. Send verification email for updated email address
3. Update customer profile and addresses

### Transition Mode:

1. Login on SFCC form and silently register in Reachfive
2. On first login after silent register, you can use the Reachfive login component
3. Reset Password and update profile allow a double update in SFCC and in Reachfive
4. Send verification email for updated email address
5. Update customer profile and addresses

The Customer Registration and login through ReachFive works whether the Customer logs in / registers while in the checkout process or in the standard account registration / login page.

## 2.3 Limitations, Constraints

There is currently no limitations nor constraints identified with this component at this time.

## 2.4 Compatibility

The latest release is compatible with SFCC 18.3.

## 2.5 Privacy

The int\_reachfive\_sg cartridge access the SFCC Customer.ID and the SFCC Site.ID properties values and stores this information within the ReachFive platform. Moreover, during the social connect process, ReachFive stores the social data (coming from the social provider), that the Customer has willingly accepted to share, and makes this data available to the merchant.

During the registration / login process :

If the data is available through the social provider and if there is no previous value stored within SFCC for these properties, the following properties are set on the Customer profile on the SFCC side:

* Profile.lastName
* Profile.firstName
* Profile.email
* Profile.birthday

The following Credentials properties will be set during the registration / login process through ReachFive :

* Credentials. authenticationProviderID
  + The value set is a custom value chosen by the merchant and manageable in the ReachFive Site Preferences of the SFCC Business Manager.
* Credentials.externalID
  + This property stores the ReachFive user ID.

## 2.6 Security

The social login process has to happen on the SFCC side, which implies a risk that a malicious attacker will attempt to tamper with the data sent from the client to the server. In the next paragraph is the explanation of the authentication process and how this prevents from security issues

### Step 1: User authentication process

The user authenticates via email/password or social login. This process must be at least partially implemented using the suitable ReachFive’s SDK depending on the client platform (Web, mobile…).

ReachFive’s Identity SDKs allow different authentication flows. For server-side authentication, you must use a code response type, and set your login callback URL in the redirectUri attribute **(The URI must be whitelisted in the “Allowed Callback URLs” field of your ReachFive’s account settings).**

For security reasons, you have to list all callback URLs in the Allowed Callback URLs field of your ReachFive console.

You can find below an example using the Identity SDK for Web.

sdkUiClient.showAuth({

container: 'auth-container',

auth: {

responseType: 'code', // This is the default value when "redirectUri" attribute is set.

redirectUri: 'https://www.example.com/login/callback'

}

});

### Step 2: Handle the Authorization Response

After the authentication process, ReachFive responds to your application by redirecting the user to the callback URL specified previously (redirectUri attribute).

If the authentication succeeded, then the response contains an authorization code. If not (e.g. when the user does not approve access to his social data), the response contains an error message. The authorization code or error message that is returned to your web server appears in the query string, as shown below.

An authorization code response:

https://www.example.com/login/callback?code=A8sLD49d-lPcKyUwBaSm4oThfjp4

An error response:

https://www.example.com/login/callback?error=access\_denied

### Step 3: Exchange authorization code for ID token

After the web server receives the authorization code, it can exchange the authorization code for an ID token using an HTTP request to the ReachFive’s token endpoint.

The string YOUR\_REACHFIVE\_DOMAIN should be replaced with the domain of your ReachFive account and the string YOUR\_REACHFIVE\_CLIENT\_ID should be replaced with your client ID of your ReachFive account.

POST /oauth/token HTTP/1.1

Host: https://YOUR\_REACHFIVE\_DOMAIN

Content-Type: application/x-www-form-urlencoded

code=A8sLD49d-lPcKyUwBaSm4oThfjp4

&client\_id=YOUR\_REACHFIVE\_CLIENT\_ID

&redirect\_uri=https://www.example.com/login/callback

&grant\_type=authorization\_code

Success response example:

{

"id\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1N...",

"access\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIU...",

"token\_type": "Bearer",

"expires\_in": 3600

}

### Step 4: Retrieve user’s profile data from ID Token

After your application obtains an id\_token, you can use the token to retrieve the user’s profile data by parsing it. An id\_token is a JWT (JSON Web Token), that is, a cryptographically signed Base64-encoded JSON object. As you are communicating directly with ReachFive over HTTPS channel, it is normally not critical that you validate your id\_token before parsing it. However most API libraries combine the validation with the work of decoding the base64 and parsing the JSON.

Example:

var decodedToken = jwt.verify(idToken, clientSecret);

Success example:

{

"sub": "AVPw-jHcQG5c\_BvJk9e\_",

"given\_name": "John",

"family\_name": "Doe",

"email": "john.doe@example.com"

}

# 3. Implementation Guide

## 3.1 Setup

### 3.1.1 Import Cartridge

Import the **int\_reachfive** and **int\_reachfive\_sg** cartridge into the UXStudio Workspace.

* Open UXStudio
* Click File -> Import -> General -> Existing Projects Into Workspace
* Browse to the directory where you saved the "**:int\_reachfive\_sg:int\_reachfive**" cartridge.
* Click Finish.
* Click OK when prompted to link the cartridge to the sandbox.

### 3.1.2 Import Site Preferences and Custom Objects

Import the system object customizations into the SFCC Business Manager.

* Log into the SFCC Business Manager.
* Click Administration -> Sites Development -> Site Import & Export
* Click the upload link or button in the “Import” section.
* Use the upload control to browse for the site\_template.zip file
* Note: if you are configuring the SiteGenesis site, you can import the site\_template.zip like this, otherwise you should follow these steps:
  + Cd in the metadata downloaded from bitbucket
  + Cd /site\_template/sites
  + Rename the folder. SiteGenesis with the Id of your site
  + zip -r site\_template.zip site\_template
* Click Upload
* Select the site\_template.zip file that was just uploaded
* Click Import

## 3.2 Configuration

### 3.2.1 Add the Reach Five Cartridge to your Storefront Cartridge Path

Append int\_reachfive\_sg to the effective cartridge path

* Log into the SFCC Business Manager.
* Click Administration -> Sites -> Manage Sites
* Select the desired site
* Click on the Settings tab.
* Append ":int\_ reachfive " to the "Cartridges" field.
* Click Apply

### 3.2.2 Configure Reach Five Site Preferences

Configure Reach Five Custom Preferences using the SFCC Business Manager

* Log into the SFCC Business Manager
* Select the desired site from the tabs across the top of the page.
* Click Site Preferences -> Custom Preferences
* Fill in the Site Preferences

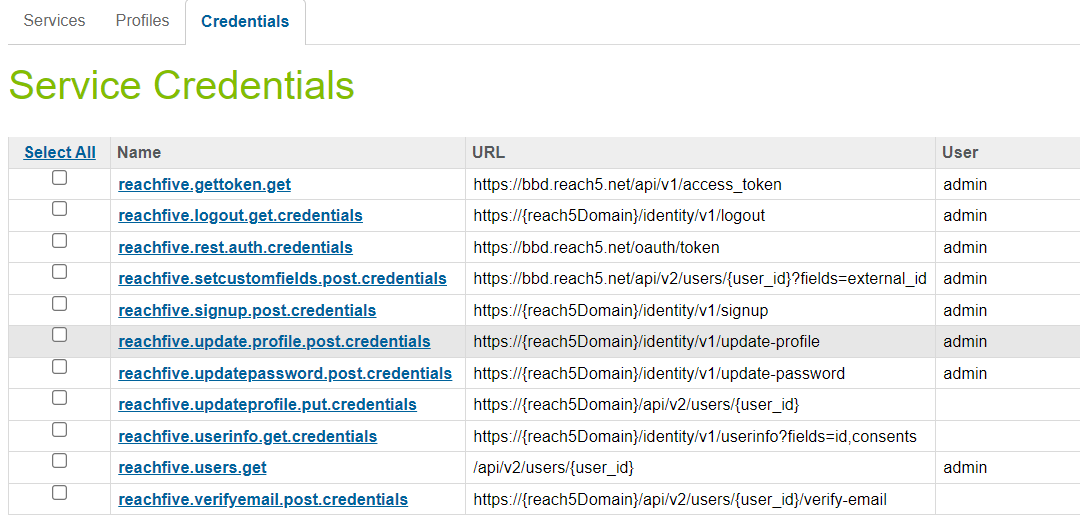
|  |  |
| --- | --- |
| Entity | Description |
| cartridgeControllersName | Name of the cartridge where are controllers (i.e. app\_storefront\_controllers) |
| isReachFiveEnabled | If set to true, ReachFive is activated |
| isReach5ThemeActive | If set to true, default theme is activated (actually is not used currently) |
| reach5Domain | The domain name of the ReachFive environment |
| reach5ApiKey | The API key required to access the ReachFive Environment |
| reach5ClientSecret | The client secret key required to access the ReachFive environment |
| reachFiveProviderId | The name of the provider that will be used to populate the Credentials.authenticationProviderID properties value for the Customers who will register or log in on the SFCC storefront through ReachFive |
| isReachFastRegister | If “No”, pending the Social Login or SSO, if the user profile does not exist in SFCC database, the user is redirected to a registration page, with pre-filled but editable fields. If “Yes”, pending the Social Login or SSO, if the user profile does not exist in SFCC Database, it is created using the information from the Social Login or SSO. |
| isReachFiveLoginAllowed | If “No” the Social Login scenario is enabled: the social button are added to login page, registration page, checkout page, personal data page.  If “Yes” the Single Sign On scenario is enable: the reach five login / signup widget will be present in login page, registration page, checkout page. In Personal Data page, there will be the social account widget that permits to the customer to enable or disable social account related to his profile. |
| reach5ManagementScope | Space-delimited list of Management permissions. The default value is “manage:users read:users” |
| reach5ProfileFieldsJSON | ReachFive JSON allows the flexible configuration for updating profiles fields without code changing. It determines which fields need to be synched and set mapping between SFCC and ReachFive profile fields. It consists of such Objects: profile, address, consents, custom\_fields. Each Object matches to appropriate ReachFive Profile Object.  Here are used key-value pairs to map SFCC and ReachFive fields. Key is SFCC profile attribute and value is ReachFive field. Also is possible to use custom Profile attribute in key. It should contain ‘custom.’ prefix.  To send new attribute you need to add it to appropriate Object in such format:  (custom.)SfccAttribute: reach\_five\_field  For ‘address’ JSON Object are used SFCC  CustomerAddress attributes for other  Objects are used SFCC Profile attributes  JSON Example:  {  "profile": {  "firstName": "given\_name",  "lastName": "family\_name",  "birthday": "birthdate",  "gender": "gender",  "companyName": "company",  "phoneHome": "phone\_number"  },  "address": {  "ID": "title",  "fullName": "recipient",  "address1": "street\_address",  "city": "locality",  "postalCode": "postal\_code",  "stateCode": "region",  "countryCode": "country",  "phone": "phone\_number"  },  "consents": {  "custom.isNewsletter": "newsletter"  },  "custom\_fields": {  "custom.cardNo": "loyalty\_card\_number"  }  } |
| isReachFiveTransitionActive | Enable the transition state of the Reach Five integration. It means that customers will login on SFCC form first (this login will create a new Reachfive profile and link it to customer profile).  After successful login, next time a customer will have ReachFive login form on the same device. |
| reachFiveTransitionCookieDuration | Duration of the "Transition" cookie in days. Required to identify the user already migrated to ReachFive. It means how long the ReachFive login form will automatically display for this device during this option enable. |
| reachFiveLoginCookieDuration | Duration of the “Login” cookie in days. It defines how much time you'll check the reachfive cookie to login automatically again. |
| isReachFiveSessionForcedAuth | Automatic user authentication from any page. This enables automatic authentication from any page with a reachfive long session cookie. |

### 3.2.3 Configure Reach Five Services

Services should be imported from site\_template.zip with other metadata.

Configure services credentials:

1. Log into the SFCC Business Manager.
2. Navigate to Administration -> Operations -> Services
3. Click on the Credentials tab.
4. If Credentials URL contains ‘demandware.og4.me’ domain then replace it to current reach5Domain. If URL contains ‘{reach5Domain}’ domain then it will be replaced automatically with ‘reach5Domain’ Site Preference value.

Configure services logging.

1. Log into the SFCC Business Manager.
2. Navigate to Administration -> Operations -> Services
3. Click on the Services tab and open needed service.
4. Set ‘Communication Log Enabled’ flag
5. Set ‘Log Name Prefix’ to ‘reachfive’. The service responses and requests will be logged in the files with ‘service-reachfive-’ prefix. To see them navigate to Business Manager -> Administration -> Site Development -> Development Setup -> Security Log Files

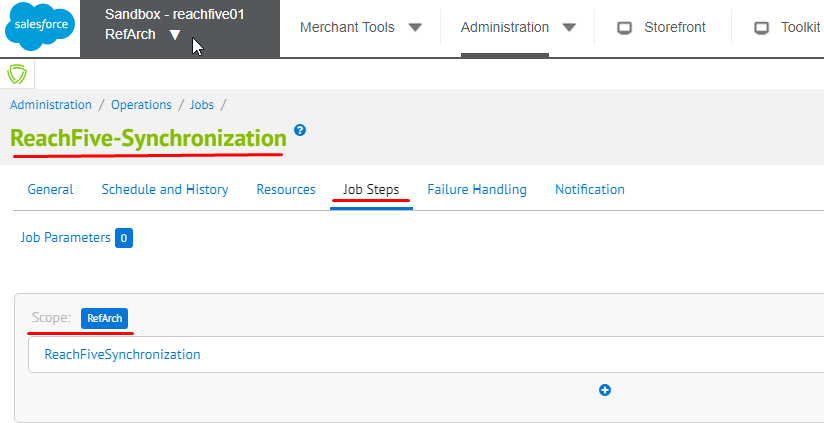
### 3.2.4 Configure Reach Five Jobs

Jobs should be imported from site\_template.zip with other metadata.

Here is used ‘ReachFive-Synchronization’ job wich synchronizes prifile data from SFCC to ReachFive.

### Set Scope:

1. Log into the SFCC Business Manager.
2. Navigate to Administration -> Operations -> Jobs
3. Click on the ‘Job Steps’ tab.
4. Set needed site in the Scope



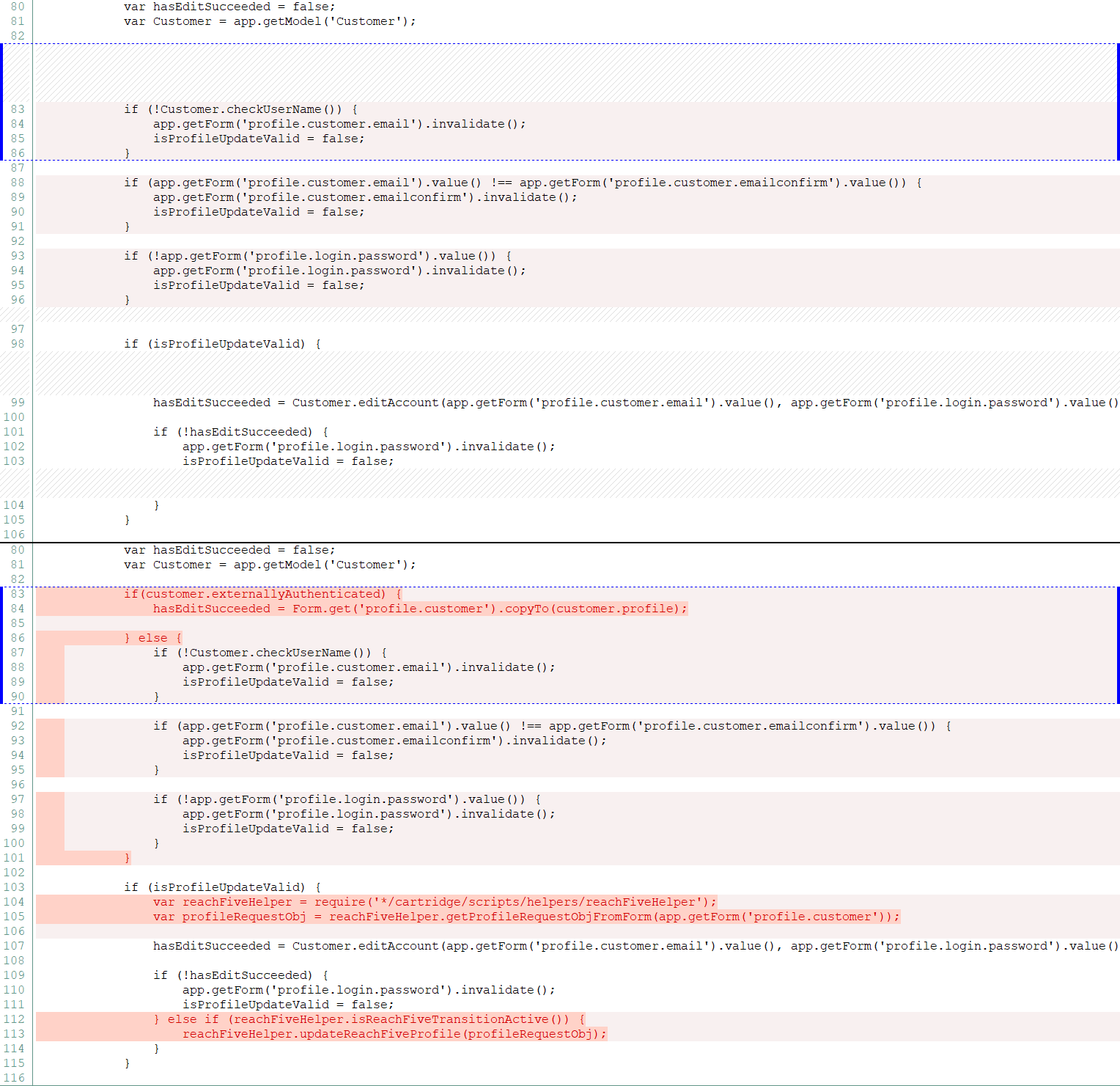
### Job Schedule (Job is scheduled to run every 5 minutes by default):

1. Log into the SFCC Business Manager.
2. Navigate to Administration -> Operations -> Jobs
3. Click on the ‘Schedule and History’ tab.
4. Set needed schedule or disable it

## 3.3 Custom Code

### 3.3.1 Custom Code. Controllers.

app\_storefront\_controllers/cartridge/controllers/Account.js

The code to add is:

—------------

if(customer.externallyAuthenticated) {

hasEditSucceeded = Form.get('profile.customer').copyTo(customer.profile);

} else {

—------------

}

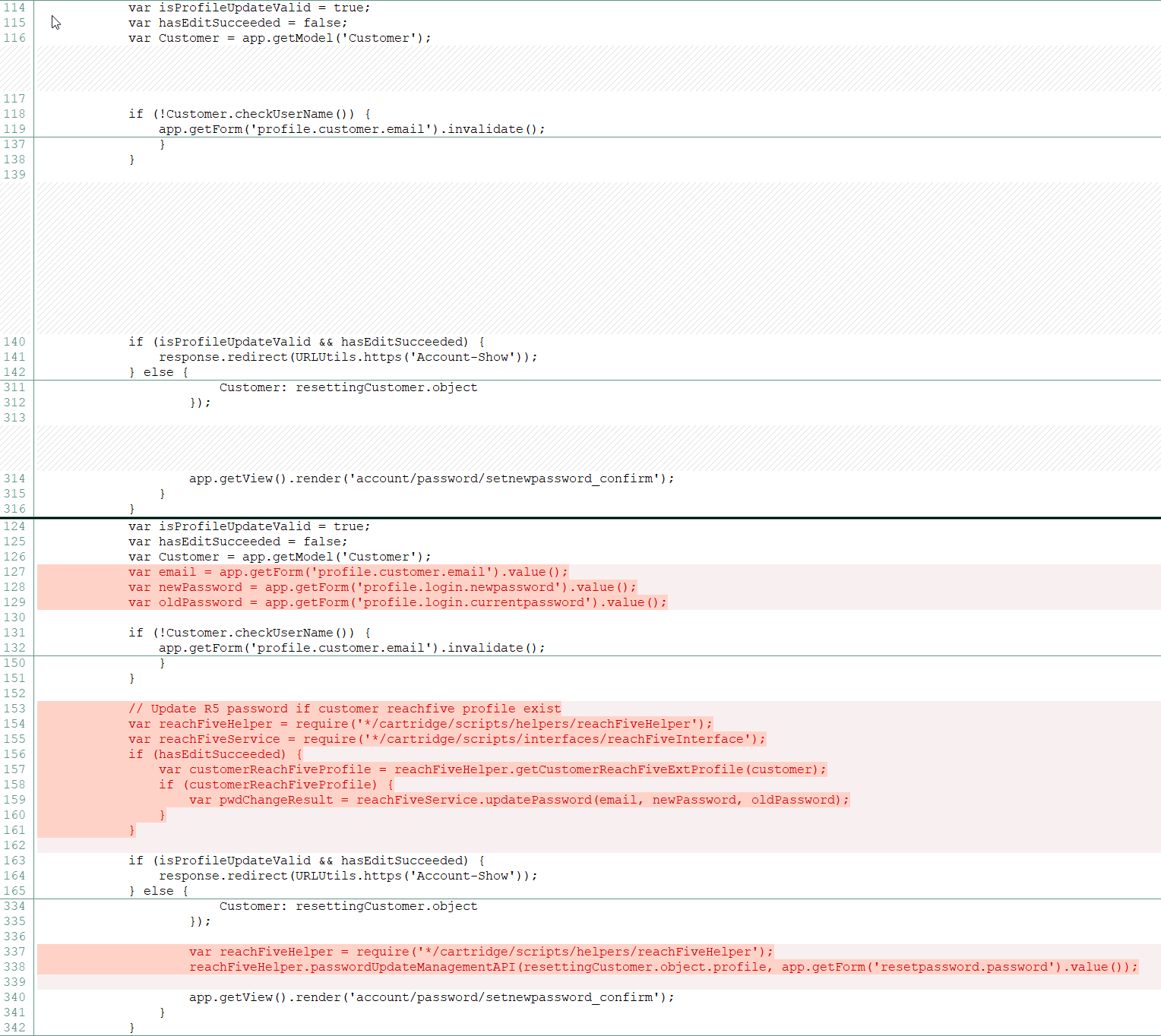
—------------

var reachFiveHelper = require('\*/cartridge/scripts/helpers/reachFiveHelper');

var profileRequestObj = reachFiveHelper.getProfileRequestObjFromForm(app.getForm('profile.customer')); —------------

} else if (reachFiveHelper.isReachFiveTransitionActive()) {

reachFiveHelper.updateReachFiveProfile(profileRequestObj);

The code to add is:

—------------

var email = app.getForm('profile.customer.email').value();

var newPassword = app.getForm('profile.login.newpassword').value();

var oldPassword = app.getForm('profile.login.currentpassword').value();

—------------

// Update R5 password if customer reachfive profile exist

var reachFiveHelper = require('\*/cartridge/scripts/helpers/reachFiveHelper');

var reachFiveService = require('\*/cartridge/scripts/interfaces/reachFiveInterface');

if (hasEditSucceeded) {

var customerReachFiveProfile = reachFiveHelper.getCustomerReachFiveExtProfile(customer);

if (customerReachFiveProfile) {

var pwdChangeResult = reachFiveService.updatePassword(email, newPassword, oldPassword);

}

}

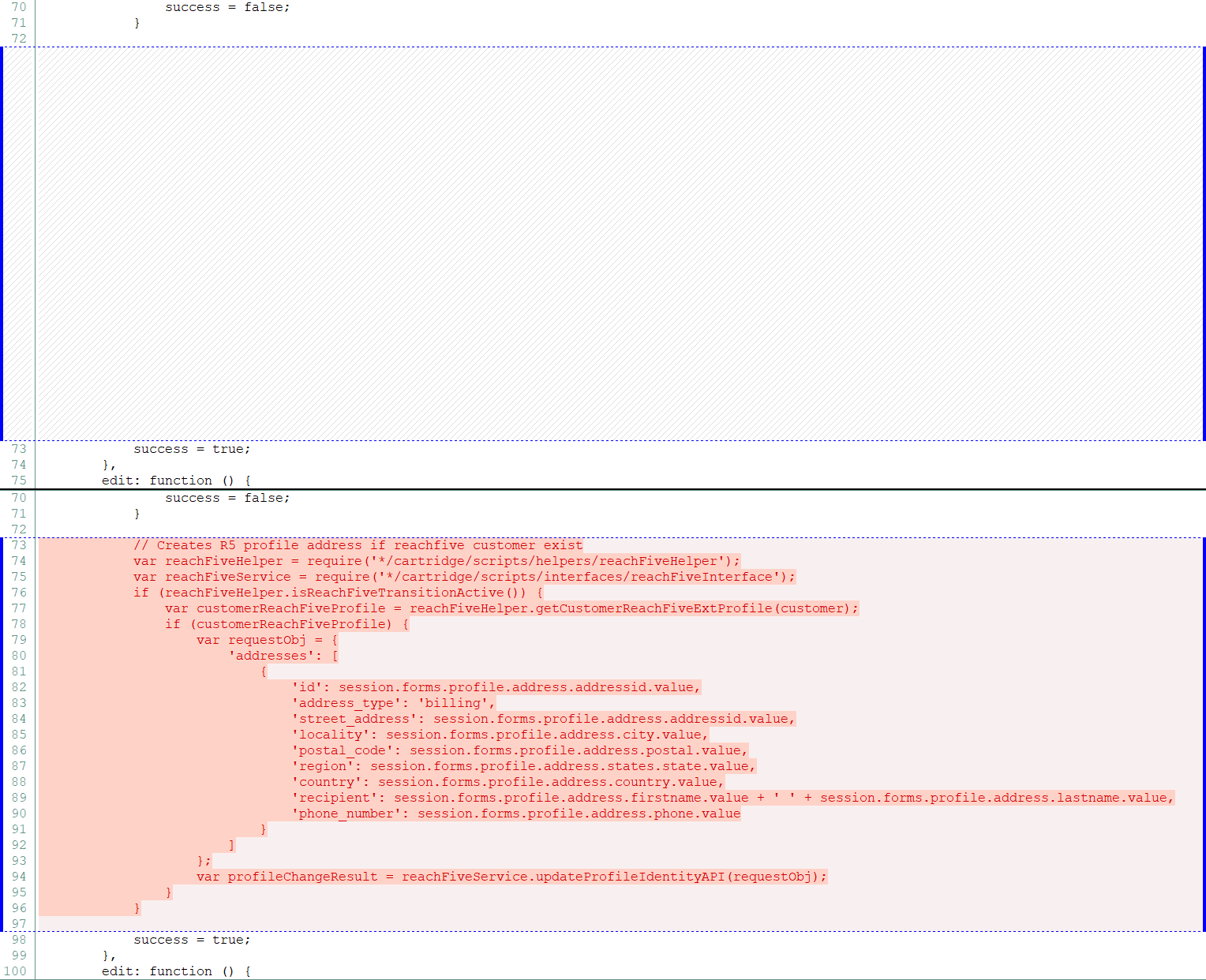
—------------

var reachFiveHelper = require('\*/cartridge/scripts/helpers/reachFiveHelper');

reachFiveHelper.passwordUpdateManagementAPI(resettingCustomer.object.profile,

app.getForm('resetpassword.password').value());

app\_storefront\_controllers/cartridge/controllers/Address.js



The code to add is:

—------------

// Creates R5 profile address if reachfive customer exist

var reachFiveHelper = require('\*/cartridge/scripts/helpers/reachFiveHelper');

var reachFiveService = require('\*/cartridge/scripts/interfaces/reachFiveInterface');

if (reachFiveHelper.isReachFiveTransitionActive()) {

var customerReachFiveProfile = reachFiveHelper.getCustomerReachFiveExtProfile(customer);

if (customerReachFiveProfile) {

var requestObj = {

'addresses': [

{

'id': session.forms.profile.address.addressid.value,

'address\_type': 'billing',

'street\_address': session.forms.profile.address.addressid.value,

'locality': session.forms.profile.address.city.value,

'postal\_code': session.forms.profile.address.postal.value,

'region': session.forms.profile.address.states.state.value,

'country': session.forms.profile.address.country.value,

'recipient': session.forms.profile.address.firstname.value + ' ' +

session.forms.profile.address.lastname.value,

'phone\_number': session.forms.profile.address.phone.value

}

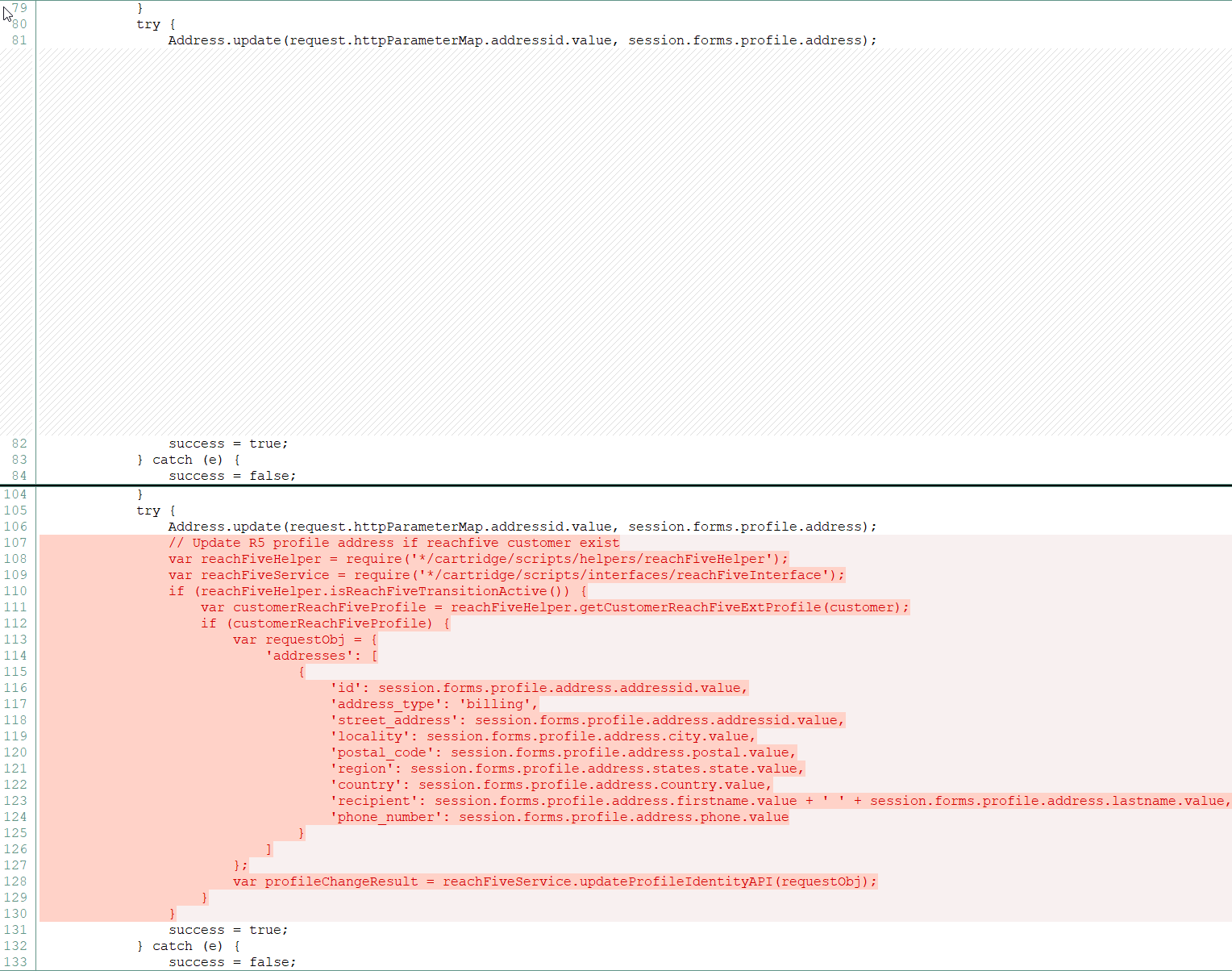
]

};

var profileChangeResult = reachFiveService.updateProfileIdentityAPI(requestObj);

}

}



The code to add is:

—------------

// Update R5 profile address if reachfive customer exist

var reachFiveHelper = require('\*/cartridge/scripts/helpers/reachFiveHelper');

var reachFiveService = require('\*/cartridge/scripts/interfaces/reachFiveInterface');

if (reachFiveHelper.isReachFiveTransitionActive()) {

var customerReachFiveProfile = reachFiveHelper.getCustomerReachFiveExtProfile(customer);

if (customerReachFiveProfile) {

var requestObj = {

'addresses': [

{

'id': session.forms.profile.address.addressid.value,

'address\_type': 'billing',

'street\_address': session.forms.profile.address.addressid.value,

'locality': session.forms.profile.address.city.value,

'postal\_code': session.forms.profile.address.postal.value,

'region': session.forms.profile.address.states.state.value,

'country': session.forms.profile.address.country.value,

'recipient': session.forms.profile.address.firstname.value + ' ' + session.forms.profile.address.lastname.value,

'phone\_number': session.forms.profile.address.phone.value

}

]

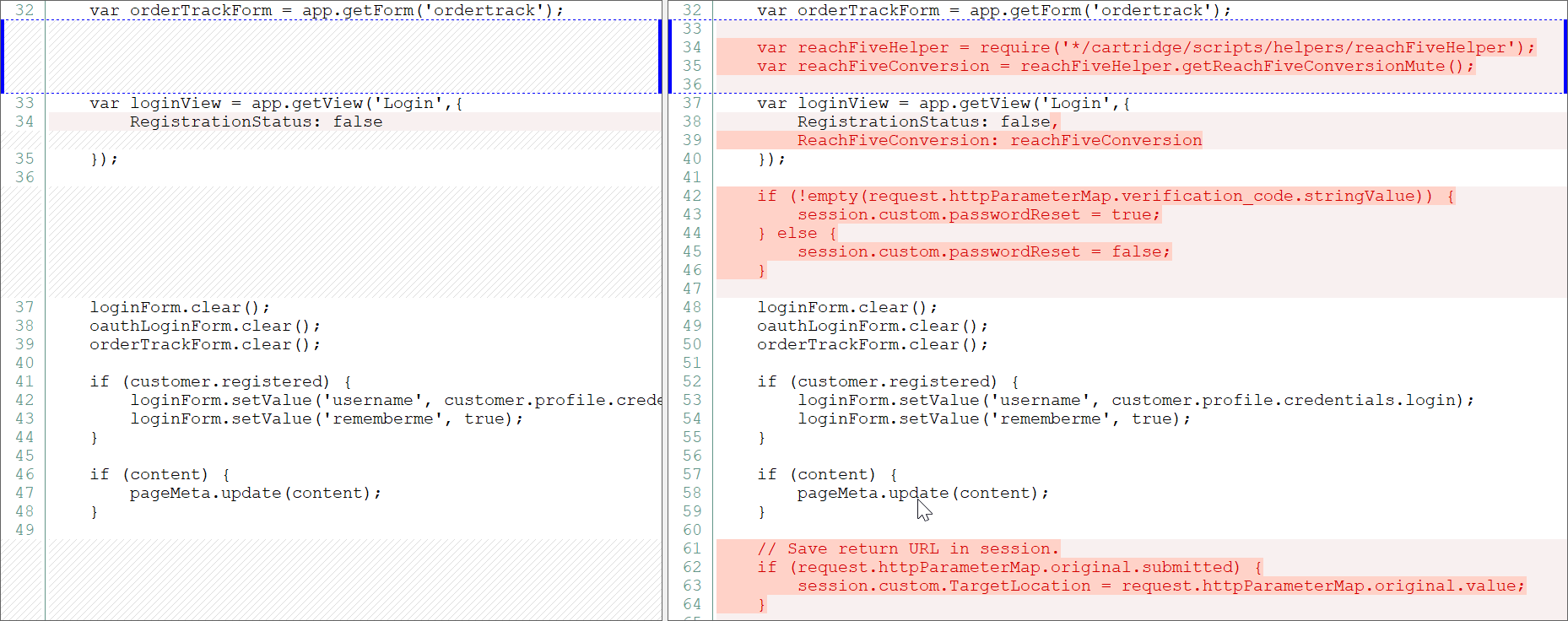
};

var profileChangeResult = reachFiveService.updateProfileIdentityAPI(requestObj);

}

}

app\_storefront\_controllers/cartridge/controllers/Login.js



The code to add is:

—------------

var reachFiveHelper = require('\*/cartridge/scripts/helpers/reachFiveHelper');

var reachFiveConversion = reachFiveHelper.getReachFiveConversionMute();

—------------

ReachFiveConversion: reachFiveConversion

—------------

if (!empty(request.httpParameterMap.verification\_code.stringValue)) {

session.custom.passwordReset = true;

} else {

session.custom.passwordReset = false;

}

—------------

// Save return URL in session.

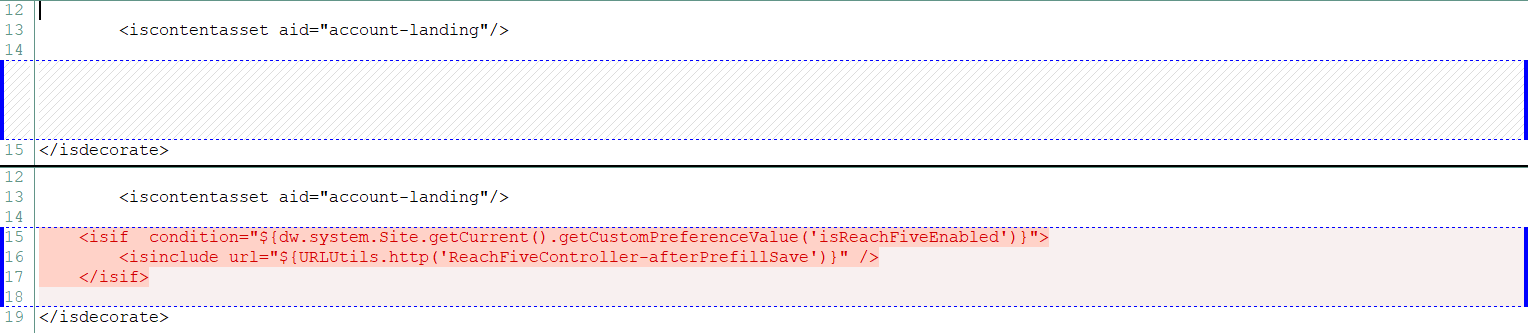
if (request.httpParameterMap.original.submitted) {

session.custom.TargetLocation = request.httpParameterMap.original.value;

}

### 3.3.2 Custom Code. Templates.

app\_storefront\_core/cartridge/templates/default/account/accountoverview.isml



The code to add is:

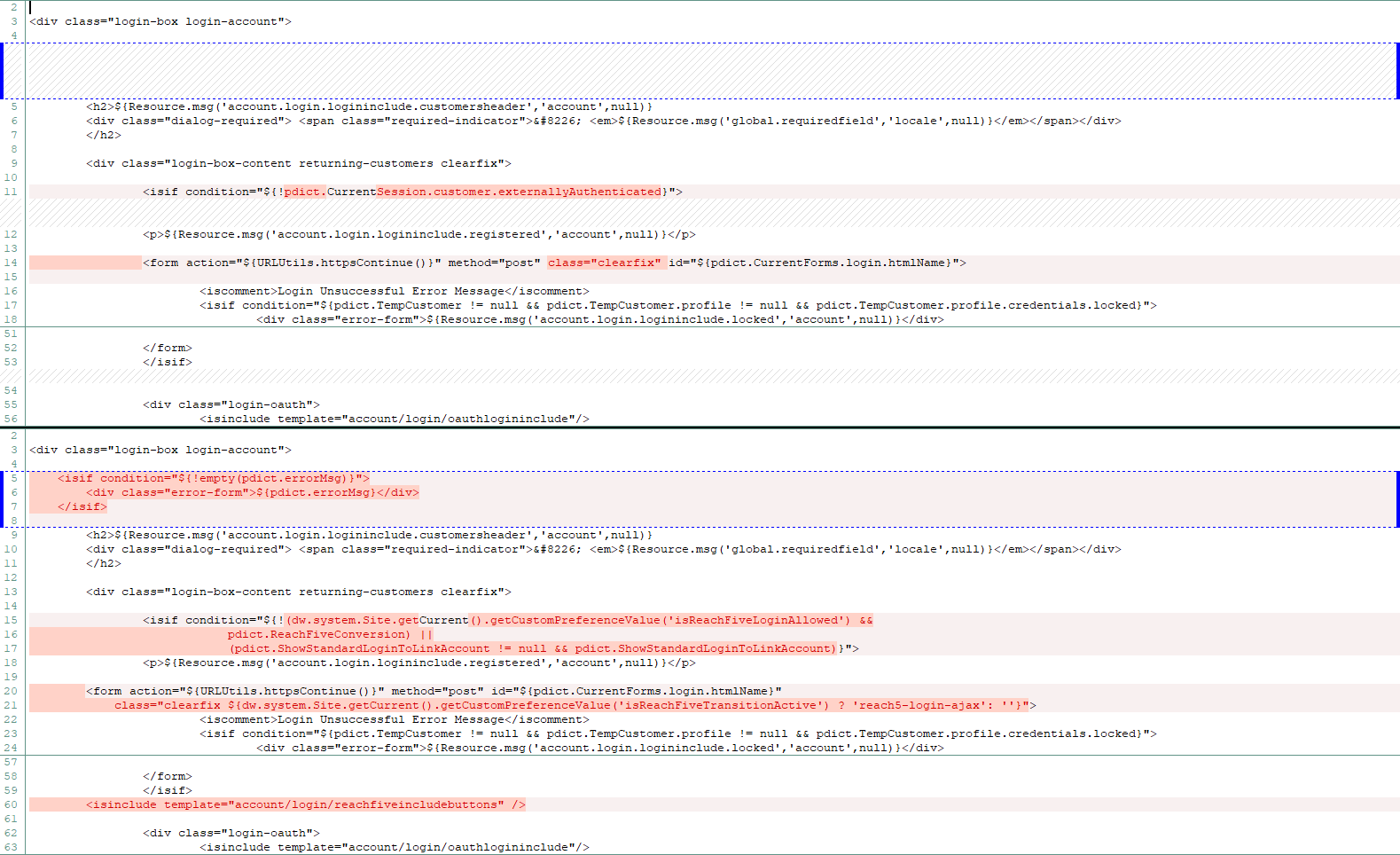
—------------

<isif condition="${dw.system.Site.getCurrent().getCustomPreferenceValue('isReachFiveEnabled')}">

<isinclude url="${URLUtils.http('ReachFiveController-afterPrefillSave')}" />

</isif>

app\_storefront\_core/cartridge/templates/default/account/login/logininclude.isml



The code to add is:

—------------

<isif condition="${!empty(pdict.errorMsg)}">

<div class="error-form">${pdict.errorMsg}</div>

</isif>

—------------

<isif condition="${!(dw.system.Site.getCurrent().getCustomPreferenceValue('isReachFiveLoginAllowed') && pdict.ReachFiveConversion) ||

(pdict.ShowStandardLoginToLinkAccount != null && pdict.ShowStandardLoginToLinkAccount)}">

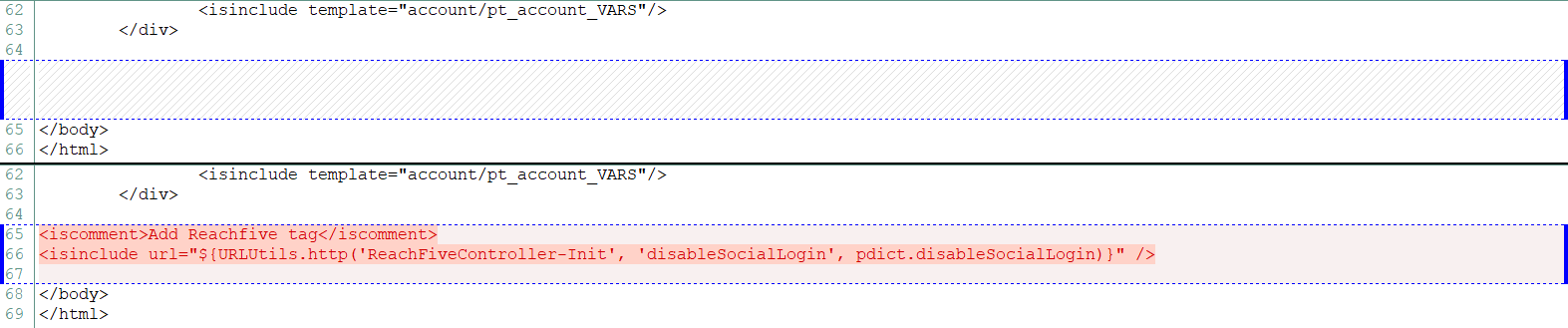
—------------

<form action="${URLUtils.httpsContinue()}" method="post" id="${pdict.CurrentForms.login.htmlName}" class="clearfix ${dw.system.Site.getCurrent().getCustomPreferenceValue('isReachFiveTransitionActive') ? 'reach5-login-ajax': ''}">

—------------

<isinclude template="account/login/reachfiveincludebuttons" />

app\_storefront\_core/cartridge/templates/default/account/pt\_account.isml

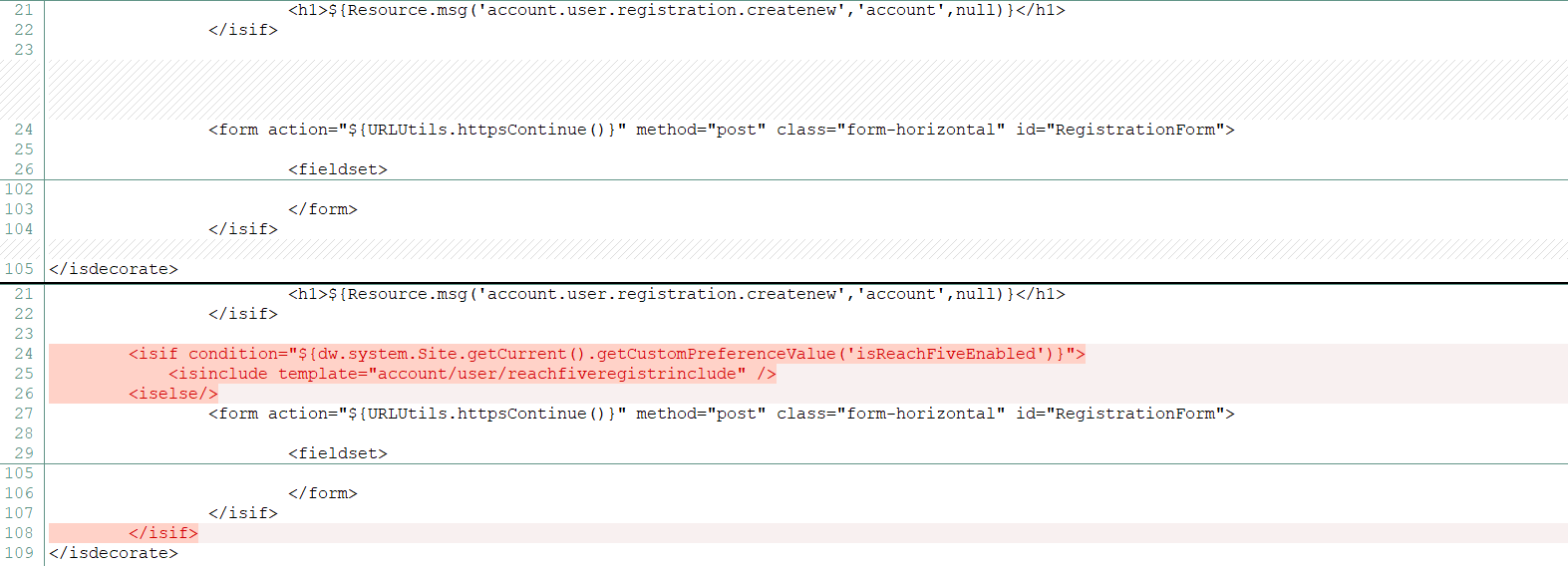
The code to add is:

—------------

<iscomment>Add Reachfive tag</iscomment>

<isinclude url="${URLUtils.http('ReachFiveController-Init', 'disableSocialLogin', pdict.disableSocialLogin)}" />

app\_storefront\_core/cartridge/templates/default/account/user/registration.isml

The code to add is:

—------------

<isif condition="${dw.system.Site.getCurrent().getCustomPreferenceValue('isReachFiveEnabled')}">

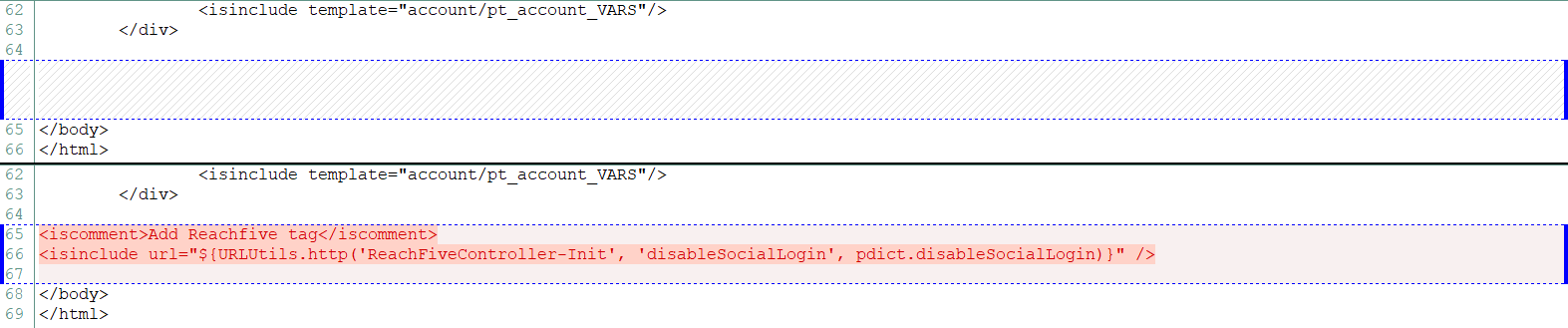
<isinclude template="account/user/reachfiveregistrinclude" />

<iselse/>

—------------

</isif>

app\_storefront\_core/cartridge/templates/default/checkout/cart/pt\_cart.isml

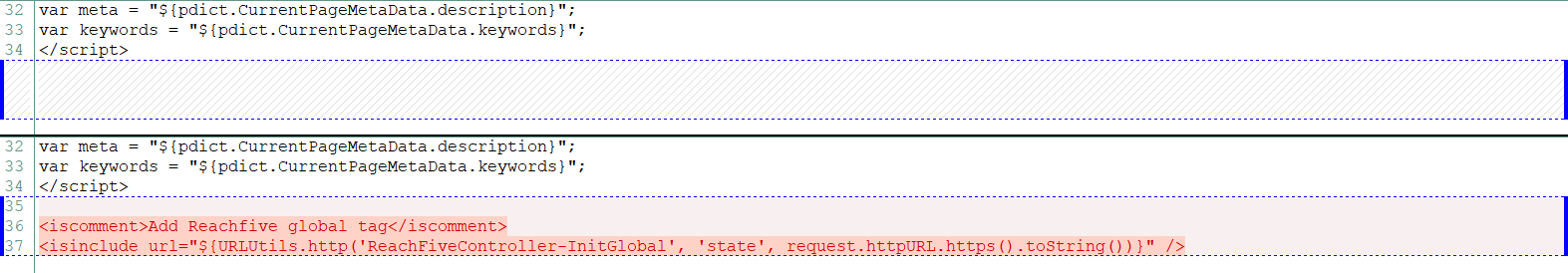
The code to add is:

—------------

<iscomment>Add Reachfive tag</iscomment>

<isinclude url="${URLUtils.http('ReachFiveController-Init', 'cart', '1')}" />

app\_storefront\_core/cartridge/templates/default/components/footer/footer\_UI.isml

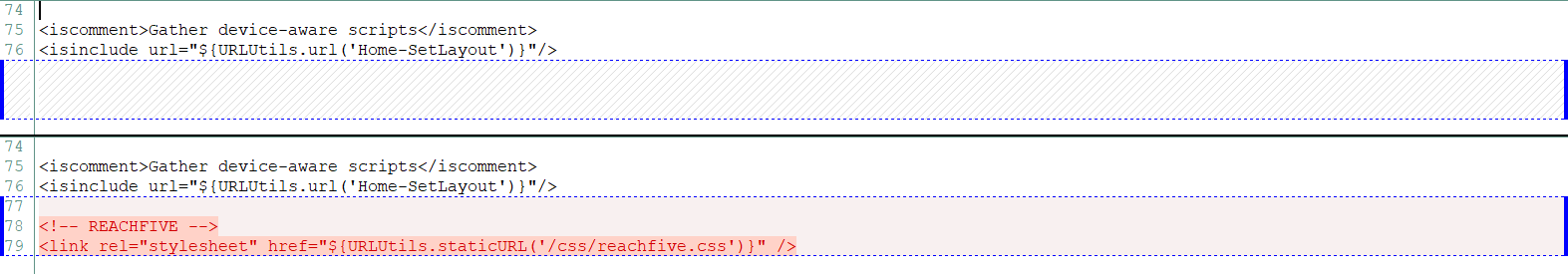
The code to add is:

—------------

<iscomment>Add Reachfive global tag</iscomment>

<isinclude url="${URLUtils.http('ReachFiveController-InitGlobal', 'state', request.httpURL.https().toString())}" />

app\_storefront\_core/cartridge/templates/default/components/header/htmlhead.isml

The code to add is:

—------------

<!-- REACHFIVE -->

<link rel="stylesheet" href="${URLUtils.staticURL('/css/reachfive.css')}" />

The Int\_reachfive\_sg cartridge includes some english written text that may need to be translated depending on the storefront language. The text elements are stored in the reachfive.properties file.

## 3.4 External Interfaces

A couple of services will be called during the social registration / login process. The first call generates a token which is required by the second call.

### Authentication

In order to access to the API, you will need to provide an access token to authenticate with the API server. That token will be required for all API requests.

You can acquire that token with the API endpoint described in the following section. Once you have acquired the API token, it may be provided preferably via an HTTP header.

### Get an access token

Example Request

POST /oauth/token HTTP/1.1

Host: https://YOUR\_DOMAIN

Content-Type: application/json

{

"grant\_type": "client\_credentials",

"client\_id": "YOUR\_CLIENT\_ID",

"client\_secret": "YOUR\_CLIENT\_SECRET",

"scope": "read:users manage:users"

}

Example Response

{

"access\_token": "kGu...uLs",

"expires\_in": 86400,

"token\_type": "Bearer"

}

### Update User

Updates the specified user by setting the values of the object’s properties passed. Any root properties (or custom fields) not provided will be left unchanged.

These are the user’s attributes that can be updated at the root level:

external\_id

email

email\_verified

phone\_number

custom\_fields

given\_name

middle\_name

family\_name

name

nickname

username

birthdate

gender

address

phone\_number

picture

company

custom\_fields

Example Request

PUT /api/v2/users/AVqvOB58Fg6nZfQ0ZqXt?fields=id,name,email,birthdate HTTP/1.1

Host: https://YOUR\_DOMAIN

Authorization: Bearer YOUR\_ACCESS\_TOKEN

{

"birthdate": "1981-10-13",

"nickname": "Johnny"

}

Example Response

{

"id": "AVqvOB58Fg6nZfQ0ZqXt",

"name": "John Doe",

"email": "johndoe@example.com",

"birthdate": "1981-10-13"

}

## 3.5 Firewall Requirements

No firewall changes are required.

# 4. Data Storage

## 4.1 Availability

The ReachFive platform is designed to be up at any time.

## 4.2 Support

The following individuals should be contacted in case of defect fixes or if improvements are needed for this component :

|  |  |  |
| --- | --- | --- |
| Name | Role | Email |
| Guillaume Partenet | ReachFive  Customer success Manager | guillaume@reach5.co |
| Gianluca Mirabelli | Salesforce Commerce Cloud  Technical Architect | gmirabelli@salesforce.com |
| Aristide Okalla | Salesforce Commerce Cloud  Technical Architect | aokalla@salesforce.com |

# 5. User Guide

## 5.1 Roles, Responsibilities

The roles and responsibilities are shared among ReachFive and the merchant as described in the following table:

|  |  |
| --- | --- |
| Who | Role & responsibilities |
| ReachFive | * Provide to the merchant the information required to connect to the ReachFive API : APISecret / APIKey * Provide and maintain the merchant’s ReachFive platform. |
| Merchant | * Integrate the Int\_reachfive\_sg cartridge in the code version of its site following the documentation. * Subscribe to the ReachFive service. |

## 5.2 Storefront Functionality

The Int\_reachfive\_sg cartridge generates social connect buttons on the login and registration page of the SiteGenesis based site. If Providers are configured in reachFive console then their social connect buttons will be added to the form. The following screenshot illustrates what it looks like with the default ReachFive template :

### Case SSO Screen Shot



### Register



### Case Social Login

### Login:

On the first register / login with a social connect provider, an OAuth provider popin window appears, asking for the user acceptance of data sharing.

On the acceptance of this terms, the user is logged in and then redirected to the My Account page or on the shipping address page, depending if the user were in the checkout process or not.

# 6. Known Issues

No known existing issues.

# 7. Failover and recovery process

During emergencies like platform or service down situations, merchants disable the cartridge features from custom site preferences. This will enable the users to connect or register through native SFCC functionalities.

# 8. Release History

|  |  |  |
| --- | --- | --- |
| Version | **Date** | Date Changes |
| 16.1.0 | August 4 2016 | Initial Release |
| 18.3.0 | March 2018 | ReachFive API version changed and add SSO functionalities |
| 19.1.0 | September 2019 | * Fix security issues with the use of user management scope in API requests * Splitting cartridge into a core cartridge “int\_reachfive” for core features and “int\_reachfive\_sg” |
| 20.2.0 | April 2020 | Add ReachFive Profiles Synchronization Job |
| 20.3.0 | July 2020 | Use synchronous web SDK: identity-ui@1.6.0, identity-core@1.15.0 |
| 20.4.1 | February 2022 | Transition mode added with all use cases to keep data up-to-date in both SFCC and  Reachfive |
| 20.4.2 | February 2022 | Add missing services & metaconfigurations |
| 20.4.3 | March 2022 | Handle the native "remember me" feature for the R5 long session |