SEB Web API application

Contents

[Database structure 3](#_Toc456259511)

[Tables 3](#_Toc456259512)

[Views 3](#_Toc456259513)

[First task 4](#_Toc456259514)

[Purpose 4](#_Toc456259515)

[Sending data 4](#_Toc456259516)

[Returning data 5](#_Toc456259517)

[Second task 6](#_Toc456259518)

[Purpose 6](#_Toc456259519)

[Sending data 6](#_Toc456259520)

[Returning data 7](#_Toc456259521)

# Database structure

I was using Microsoft SQL Server 2008 R2.

## Tables

In “Product” table we are saving product info:

ProductID (smallint) – unique identifier

Name (varchar(50)) – product name

Rules (varchar(255)) – product rules

In “Bundle” table we are saving bundle info:

BundleID (smallint) – unique identifier

Name (varchar(50)) – bundle name

Rules (varchar(255)) – bundle rules

Value (smallint) – bundle value

One bundle can have a lot of products so all these data we are saving in “Bundle\_Products” table. “Bundle\_products” table structure:

BundleID (smallint) – bundle unique identifier

ProductID (smallint) – product unique identifier

## Views

In view “vwBundle\_Product” we show bundle products in one list:

BundleID – bundle unique identifier

Bundle – bundle name

ProductID – product unique identifier

Product – product name

# First task

## Purpose

Return 201 Created together with a JSON representation of the recommended.

## Sending data

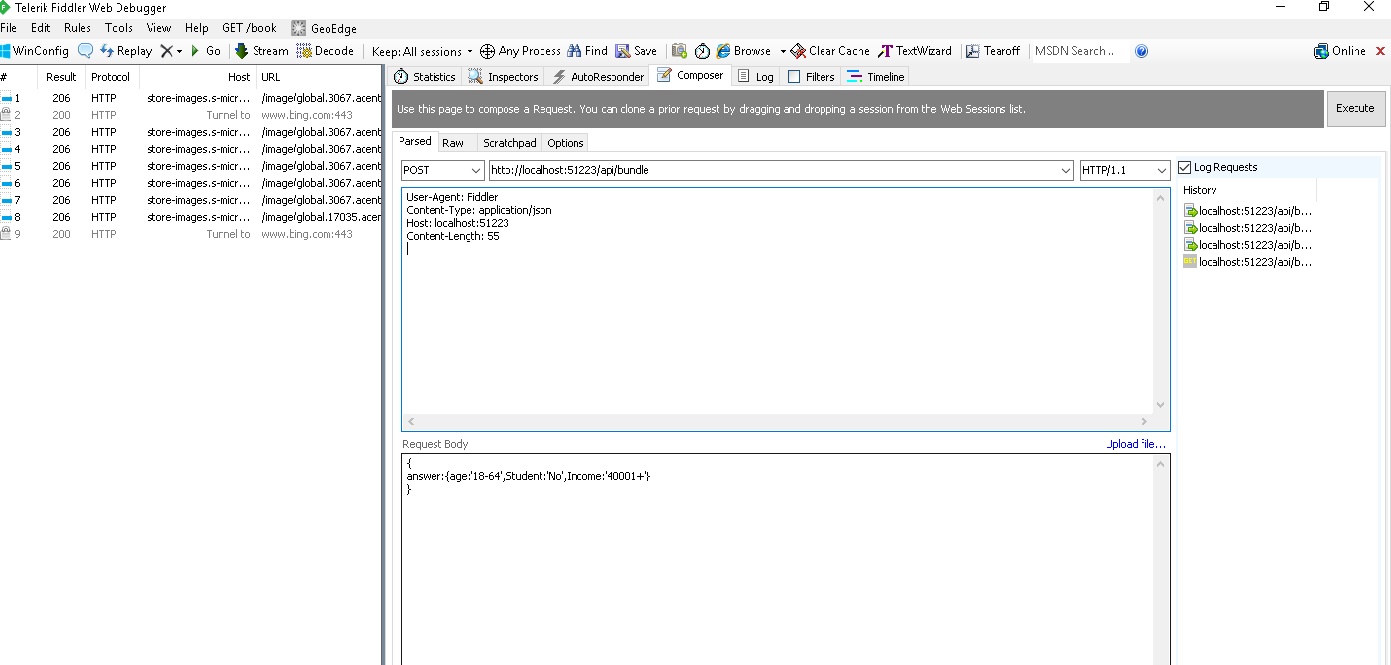
We are sending answers to the questions in JSON representation:

{

answer:{age:'18-64',Student:'No',Income:'40001+'}

}

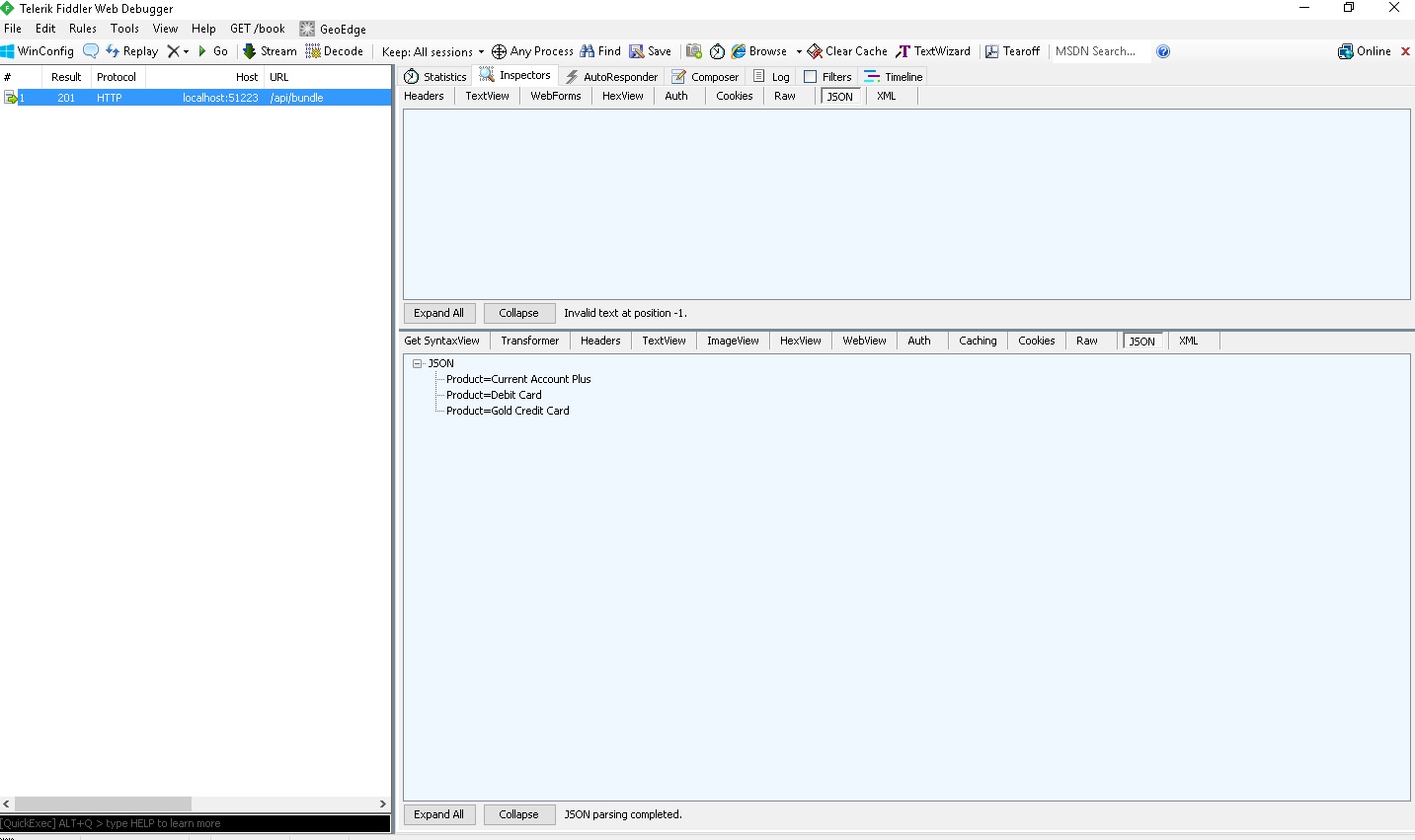
For debugging we are using “Teletrik Fiddler Web Debugger”. In picture 1 is shown data sending print screen.



Pic.1

## Returning data

After getting answers the program returns HTTP status 201 Created and recommended bundle in JSON representation picture 2.



Pic. 2

# Second task

## Purpose

Validate bundle and the answers and return HTTP 200 OK status if valid, or an appropriate HTTP response if it is not that explains why.

## Sending data

We are sending answers to the questions in JSON representation:

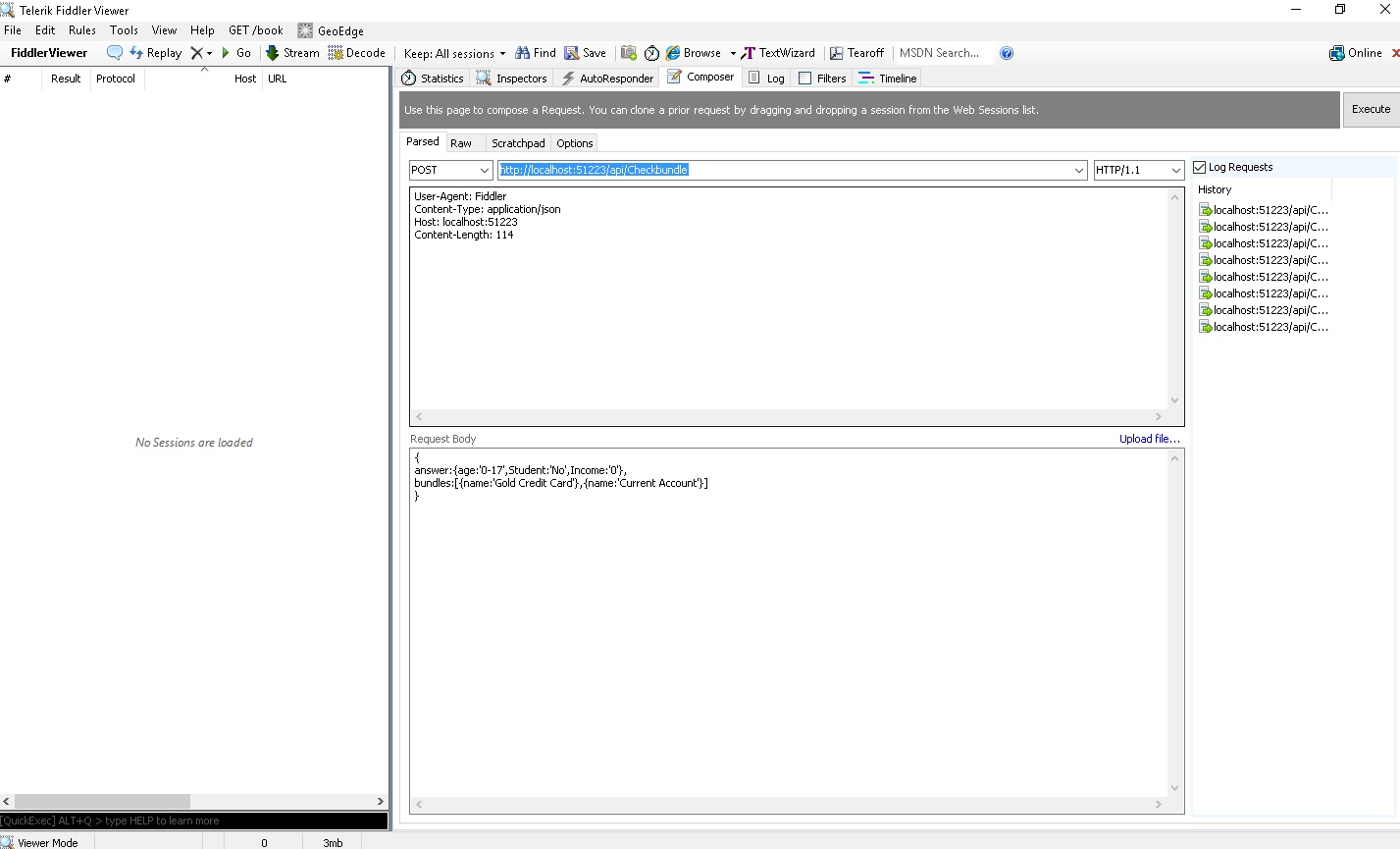
{

answer:{age:'18-64',Student:'No',Income:'40001+'},

bundles:[{name:'Gold Credit Card'},{name:'Current Account'}]

}

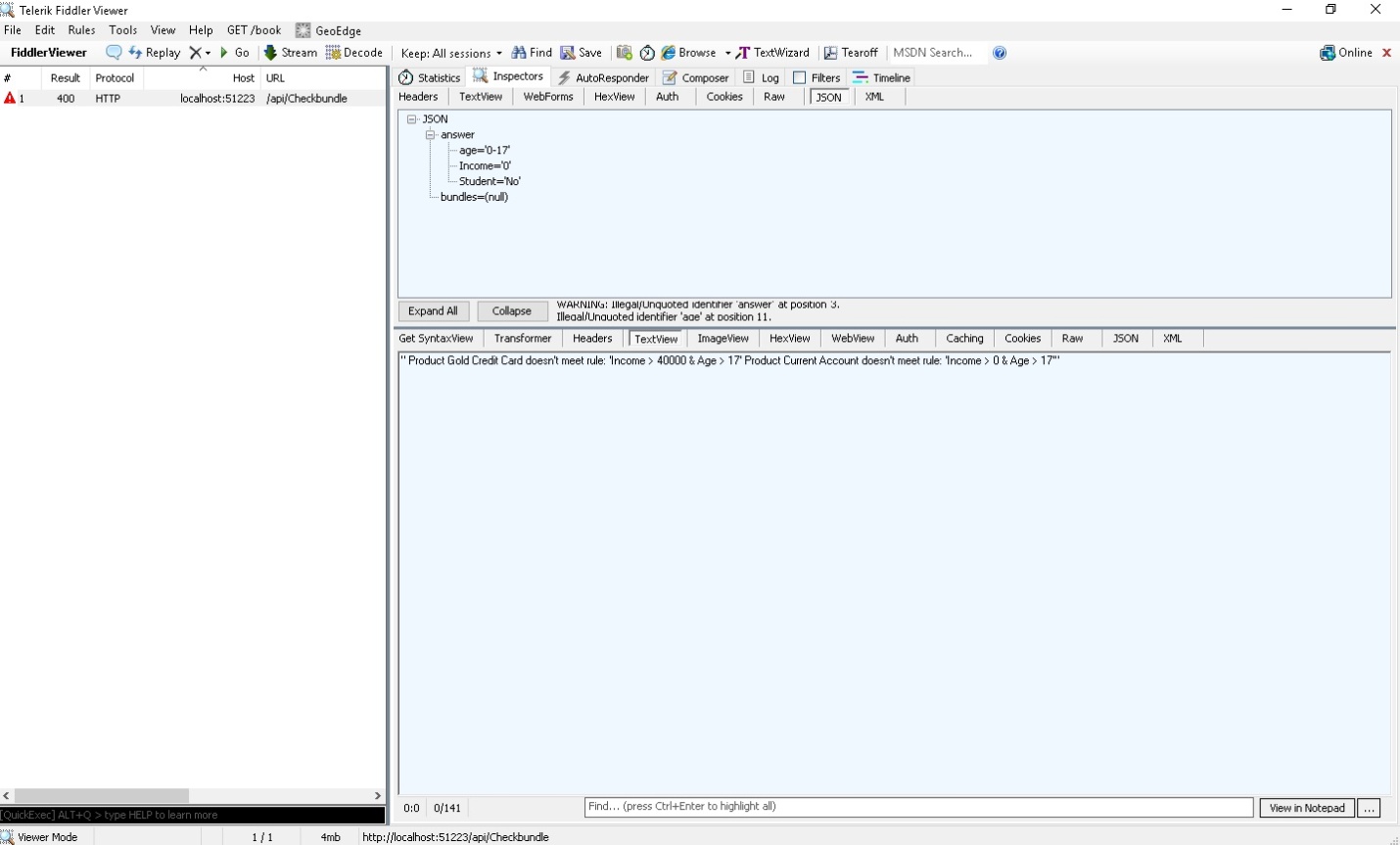
In picture 3 is shown data sending print screen.



Pic.3

## Returning data

When the received bundle products don’t meet requirements, we are returning HTTP status BAD Request and product name with rule which it should meet Pic.4, else we return status OK.



Pic. 4