WEB SERVICE SPECIFICATION

8/23/2013 Graduate Capstone

RECEIPT
REWARDS

Table of Contents

1 Introduction

This document is created to provide technical information required in making web service calls from the client applications. The scope of the document includes specification of the web service and descriptions.

1.1 Intended Audience

This documented is intended for individuals of a technical background including but not limited to developers, testers, product teams or administrators.

1.2 References

- http://www.healthit.gov/sites/default/files/nhin-web-services-registry-production-specification-v2.0-1.pdf
- http://www.fdsn.org/webservices/FDSN-WS-Specifications-1.0.pdf
- http://heimshelp.deewr.gov.au/sites/heimshelp/Support/Documents/CHESSNTechnicalSpecifications.pdf
- http://www-sk.let.uu.nl/u/d2r-6b.pdf
- http://www.civilservice.gov.uk/wp-content/uploads/2012/04/ePIMS_Benchmarking_Web_Service_Specification.pdf

1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	05/28/2013	Started the formatting and organizing of the document.	0.1
Andy Bottom	08/19/2013	Updated the web services	0.2
Andy Bottom	08/21/2013	Finished the Technical documentation of the web services	1.0

2 Technical Documentation

The following section contains a technical explanation and overview of the web services application.

2.1 Architecture Overview

The Web Services are used in the system in order to provide data and actions to the client application. The largest role that the Web Services acts as is the Data Access Layer for the system. This by doing this, all interaction with the Database is completely removed and only limited to the services.

2.2 Technical Requirements

The Web Services are developed in Java and are implemented via utilizing the Spring MVC Framework.

The Web Services follows the RESTful API. This API follows the HTTP 1.1 principals. The accessing of the Web Services relies on protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP) and Hypertext Transfer Protocol (HTTP).

The operations that are utilized in the Web Services are the following explained in the following table and how they are used in the system.

Operation	Description
GET	The operation is used for retrieving data from the database.
POST	Passes in an object in the request which is then added as a new entry in the desired table.
PUT	Performs an update of a preexisting entry in a table.
DELETE	Removes an entry from a table in the database.

2.3 HTTP Status Codes

With performing requests over HTTP, occasionally request fail due to an number of factors. When these failures occur, a status code is returned to the client application. Below is a list and description of the most common codes that are encountered when attempting connection with Web Services.

Code	Description
200	A Successful request and response.
400	Bad request due to missing data or parameter
401	Unauthorized authentication or access
500	Internal Server Error
503	The Service is temporarily unavailable or under maintenance

2.4 Security

Due to the open nature of the Web Services, security is very important. However, the extra security designs are outside the scope of the current project. However, currently future plans to implement security are created. The plan is to eventually implement HMAC Security as a design method that will secure the web services of the system.

2.5 Testing

Testing for the web services are outside the scope of the current project. However, plans to perform automated testing of the web services are in the works. The plan is to use the program SoapUI to create automated tests on each of the modules of the Data Access Layer. By creating these tests, we can ensure that all the data entering the services will pass all validations and formatting. This will create an extra benefit of data quality in the database.

3 Web Service Definitions

The following section is an exhaustive list of all the definitions of the web services. Since this is the data access layer of the system, each database has a designated service to each table. In each service, there are mapping for being able to perform all CRUD actions on the database.

3.1 Automation

The Automation Object is part of the question which holds a list of command elements.

3.1.1 Base Mappings

/Automation

3.1.2 GET Mappings

The following methods are used to simple get an Automation or List of Automation Objects.

3.1.2.1 Get by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer. Represents the Id of the desired Automation. Should Return
Request	None
Response	Automation Object

3.1.2.2 Get by Revision Id

Category	Value
Mapping	/ById/Revision/{revisionId}
URL Parameters	revisionId: Integer. Represents the Revision ID of which the automations are associated with.
Request	None
Response	List of Automation Objects

3.1.3 POST Mappings

The following mappings are used to create a new Automation entry into the database.

3.1.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None
Request	Automation Object
Response	Automation Object with new Automation Id Number

3.1.4 PUT Mappings

The following mappings are used to update an existing Automation entry in the database.

3.1.4.1 Update Automation's Command Order/Weight

Category	Value
Mapping	/Update/Weight/
URL Parameters	None
Request	Automation Object with all Command Elements Present in the correct order
Response	Automation Object with the Command Elements in the correct order.

3.1.5 DELETE Mappings

The following mappings are used to remove Automation entries from the database.

3.1.5.1 Remove by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	AutomationId: Integer; The Id of the Automation to delete
Request	None
Response	None

3.1.5.2 Remove by Automation Id

Category	Value
Mapping	/ById/Revision/{revisionId}
URL Parameters	revisionId: Integer; The Id of the Revision which all Automations to delete are associated with.
Request	None
Response	None

3.1.6 Other Mappings

3.1.6.1 Perform Automation Test

Category	Value
Mapping	/Action/Test/Revision/{revisionId}
URL Parameters	revisionId: Integer; The Id of the Revision for which the test automation should be run.
Request	None
Response	None

3.2 Automation Flow

3.2.1 Base Mappings

/AutomationFlow

3.2.2 GET Mappings

The following methods are used to simple get an Automation Flow or List of Automation Flow Objects.

3.2.2.1 Get by Automation Flow Id

Category	Value
Mapping	/ById/AutomationFlow/{automationFlowId}
URL Parameters	automationFlowId: Integer. Represents the Id of the desired Automation Flow.
Request	None
Response	Automation Flow Object

3.2.2.2 Get by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer. Represents the Automation ID of which the automation Flows are associated with.
Request	None
Response	List of Automation Flow Objects

3.2.3 POST Mappings

The following mappings are used to create a new Automation Flow entry into the database.

3.2.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None
Request	Automation Flow Object
Response	Automation Flow Object with new Automation Flow Id Number

3.2.4 PUT Mappings

None

3.2.5 DELETE Mappings

The following mappings are used to remove Automation Flow entries from the database.

3.2.5.1 Remove by Automation Flow Id

Category	Value
Mapping	/ById/AutomationFlow/{automationFlowId}
URL Parameters	automationFlowId: Integer; The Id of the Automation Flow to delete
Request	None
Response	None

3.2.5.2 Remove by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer; The Id of the Automation which all Automation Flows are associated with.
Request	None
Response	None

3.2.5.3 Remove by Question Id

Category	Value
Mapping	/ById/Question/{questionId}
URL Parameters	questionId: Integer; The Id of the Question.
Request	None
Response	None

3.3 Code

3.3.1 Base Mappings

/Code

3.3.2 GET Mappings

The following methods are used to simple get a Codes or List of Codes Objects.

3.3.2.1 Get All Codes

Category	Value
Mapping	/All/
URL Parameters	None
Request	None
Response	List of Codes

3.3.2.2 Get by Code Type ID

Category	Value
Mapping	/ById/CodeType/{codeTypeId}
URL Parameters	codeTypeId: Integer. Represents the Code Type ID that the code is associated with.
Request	None
Response	List of Code Objects

3.3.2.3 Get by Code Id

Category	Value
Mapping	/ById/Code/{codeId}
URL Parameters	codeId: Integer. Represents the Code ID of the Code
Request	None
Response	Code Object

3.3.2.4 Get by Code Id

Category	Value
Mapping	/ByCode/Code/{code}
URL Parameters	code: String. Represents the Code of the Code Object
Request	None
Response	List of Code Objects

3.3.3 POST Mappings

The following mappings are used to create a new Code entry into the database.

3.3.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None
Request	Code Object
Response	Code Object with new Code Id Number

3.3.4 PUT Mappings

The following mappings are used to update an existing Code entry in the database.

3.3.4.1 Update Code

Category	Value
Mapping	/Update/
URL Parameters	None
Request	Code Object with the Updated Values
Response	Code Object with the Updated Values

3.3.5 DELETE Mappings

3.3.5.1 Remove by Code Id

The following mappings are used to remove Code entries from the database.

Category	Value
Mapping	/ById/Code/{codeId}
URL Parameters	codeId: Integer; The Id of the Code
Request	None
Response	None

3.3.5.2 Remove by Code

Category	Value
Mapping	/ByCode/Code/{code}
URL Parameters	code: String; The Code of the Code Object
Request	None
Response	None

3.4 Code Type

3.4.1 Base Mappings

/CodeType

3.4.2 GET Mappings

The following methods are used to simple get a Code Type or List of Code Type Objects.

3.4.2.1 Get All Codes

Category	Value
Mapping	/All/
URL Parameters	None
Request	None
Response	List of Code Types

3.4.2.2 Get by Code Type ID

Category	Value
Mapping	/ById/CodeType/{codeTypeId}
URL Parameters	codeTypeId: Integer; Represents the Code Type ID;
Request	None
Response	CodenType

3.4.2.3 Get by Type

Category	Value
Mapping	/ByType/CodeType/{type}
URL Parameters	type: String; Represents the Type of the Code Type;
Request	None
Response	Code Type Object

3.4.3 POST Mappings

The following mappings are used to create a new Code Type entry into the database.

3.4.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None
Request	Code Type Object
Response	Code Type Object with new Code Type Id;

3.4.4 PUT Mappings

The following mappings are used to update an existing Code Type entry in the database.

3.4.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None
Request	Code Type Object with the Updated Values
Response	Code Type Object with the Updated Values

3.4.5 DELETE Mappings

The following mappings are used to remove Code Type entries from the database.

3.4.5.1 Remove by Code Type Id

Category	Value
Mapping	/ById/CodeType/{codeTypeId}
URL Parameters	codeTypeId: Integer; The Id of the Code Type;
Request	None
Response	None

3.5 Command Element

3.5.1 Base Mappings

/CommandElement

3.5.2 GET Mappings

The following mappings are used to simple get a Command Element or List of Command Element Objects.

3.5.2.1 Get by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; Represents the Id of the Command Element
Request	None
Response	Command Element

3.5.2.2 Get by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer; Represents the Automation ID;
Request	None
Response	List of Command Elements

3.5.3 POST Mappings

The following mappings are used to create a new Command Element entry into the database.

3.5.3.1 Create

Category	Value
Mapping	/Create/{automationId}
URL Parameters	automationId: Integer; Represents the Id of the Automation that the Command Element will be associated with;
Request	Command Element Object
Response	Command Element Object with new Command Element Id;

3.5.4 PUT Mappings

3.5.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None
Request	Command Element Object with the Updated Values
Response	Command Element Object with the Updated Values

3.5.5 DELETE Mappings

The following mappings are used to remove Command Element entries from the database.

3.5.5.1 Remove by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; The Id of the Command Element
Request	None
Response	None

3.6 Command Form Element Relation

3.6.1 Base Mappings

/CommandFormElementRelation

3.6.2 GET Mappings

The following mappings are used to simple get a Command Form Element Relation or List of Command Form Element Relation Objects.

3.6.2.1 Get by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; Represents the Id of the Command Element
Request	None
Response	List of Command Form Element Relation

3.6.2.2 Get by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; Represents the Form Element;
Request	None
Response	List of Command Form Element Relation

3.6.3 POST Mappings

The following mappings are used to create a new Command Form Element Relation entry into the database.

3.6.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None
Request	CommandFormElementRelation Object
Response	CommandFormElementRelation Object with new Command Form Element Relation Id;

3.6.4 PUT Mappings

None;

3.6.5 DELETE Mappings

The following mappings are used to remove Command Form Element Relation entries from the database.

3.6.5.1 Remove by Command Form Element Relation Id

Category	Value
Mapping	/ById/CommandFormElementRelation/{ commandFormElementRelationId }
URL Parameters	commandFormElementRelationId: Integer; The Id of the Command Form Element Relation
Request	None
Response	None

3.6.5.2 Remove by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; The Id of the Command Element
Request	None
Response	None

3.6.5.3 Remove by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; The Id of the Form Element
Request	None
Response	None

3.7 Command Identifier

3.7.1 Base Mappings

/CommandIdentifier

3.7.2 GET Mappings

The following mappings are used to simple get a Command Identifier or List of Command Identifier Objects.

3.7.2.1 Get by Command Identifier Id

Category	Value
Mapping	/ById/CommandIdentifier/{commandIdentifierId}
URL Parameters	commandIdentifierId: Integer; Represents the Id of the Command Identifier
Request	None
Response	Command Identifier

3.7.2.2 Get by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; Represents the Command Element;
Request	None
Response	List of Command Identifiers

3.7.3 POST Mappings

The following mappings are used to create a new Command Identifier entry into the database.

3.7.3.1 Create

Category	Value
Mapping	/Create/{commandElementId}
URL Parameters	commandElementId: Integer; Represents the Id of the Command Element that the Command Identifier will be associated with;
Request	Command Identifier Object
Response	Command Identifier Object with new Command Identifier Id;

3.7.4 PUT Mappings

The following mappings are used to update an existing Command Identifier entry in the database.

3.7.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Command Identifier Object
Response	Command Identifier Object;

3.7.5 DELETE Mappings

The following mappings are used to remove Command Identifiers entries from the database.

3.7.5.1 Remove by Command Identifier Id

Category	Value
Mapping	/ById/CommandIdentifier/{commandIdentifierId}
URL Parameters	commandIdentifierId: Integer; The Id of the Command Identifier;
Request	None
Response	None

3.7.5.2 Remove by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; The Id of the Command Element
Request	None
Response	None

3.8 Company

3.8.1 Base Mappings

/Company

3.8.2 GET Mappings

The following mappings are used to simple get a Company or List of Company Objects.

3.8.2.1 Get All

Category	Value
Mapping	/All
URL Parameters	None;
Request	None;
Response	List of Company Objects

3.8.2.2 Get by Company ID

Category	Value
Mapping	/ById/Company/{companyId}
URL Parameters	companyId: Integer; Represents the Company;
Request	None;
Response	Company Object

3.8.3 POST Mappings

The following mappings are used to create a new Company entry into the database.

3.8.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None;
Request	Company Object
Response	Company Object with new Company Id

3.8.4 PUT Mappings

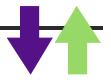The following mappings are used to update an existing Company entry in the database.

3.8.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Company Object
Response	Company Object;

3.8.5 DELETE Mappings

The following mappings are used to remove Company entries from the database.

3.8.5.1 Remove by Company ID

Category	Value
Mapping	/ById/Company/{companyId}
URL Parameters	companyId: Integer; The Id of the Company;
Request	None
Response	None

3.9 Form

3.9.1 Base Mappings

/Form

3.9.2 GET Mappings

The following mappings are used to simple get a Form or List of Form Objects.

3.9.2.1 Get by Form Id

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form;
Request	None;
Response	Form Objects;

3.9.2.2 Get by Revision ID

Category	Value
Mapping	/ById/Revision/{revisionId}
URL Parameters	revisionId: Integer; Represents the Id of the Revision;
Request	None;
Response	List of Form Objects;

3.9.3 POST Mappings

There are no creation mappings for inserting a Form. The Forms are automatically created upon a creation of a Question Object.

3.9.4 PUT Mappings

The following mappings are used to update an existing Form entry in the database.

3.9.4.1 Update

Category	Value
Mapping	/Update/Weight/
URL Parameters	None;
Request	Form Object with List of Form Elements in a correct order/weight
Response	Form Object with List of Form Elements in the new order/weight

3.9.5 DELETE Mappings

There are no mappings for deleting Form Objects. All removals are occurring automatically by the system.

3.10 Form Element Attribute
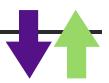
3.10.1 Base Mappings

/FormElementAttribute

3.10.2 GET Mappings

The following mappings are used to simple get a Form Element Attribute or List of Form Element Attribute Objects.

3.10.2.1 Get by Form Element Attribute Id

Category	Value
Mapping	/ById/FormElementAttribute/{formElementAttributeId}
URL Parameters	formElementAttributeId: Integer; Represents the Id of the Form Element Attribute Object;
Request	None;
Response	Form Element Attribute Object;

3.10.2.2 Get by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element;
Request	None;
Response	List of Form Element Attribute Objects

3.10.3 POST Mappings

The following mappings are used to create a new Form Element Attribute entry into the database.
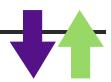
3.10.3.1 Create

Category	Value
Mapping	/Create/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element Object that the Form Element Attribute will be associated with;
Request	Form Element Attribute Object;
Response	Form Element Attribute Object with new Form Element Attribute Id

3.10.4 PUT Mappings

The following mappings are used to update an existing Form Element Attribute entry in the database.

3.10.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Form Element Attribute Object;
Response	Form Element Attribute Object;

3.10.5 DELETE Mappings

The following mappings are used to remove Form Element Attribute entries from the database.

3.10.5.1 Remove by Form Element Attribute ID

Category	Value
Mapping	/ById/FormElementAttribute/{formElementAttributeId}
URL Parameters	formElementAttributeId: Integer; The Id of the Form Element Attribute Object;
Request	None
Response	None

3.11 Form Element

3.11.1 Base Mappings

/FormElement

3.11.2 GET Mappings

The following mappings are used to simple get a Form Element or List of Form Element Objects.

3.11.2.1 Get by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element Object;
Request	None;
Response	Form Element Object;

3.11.2.2 Get by Form Id

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form;
Request	None;
Response	List of Form Element Objects;

3.11.3 POST Mappings

The following mappings are used to create a new Form Element entry into the database.
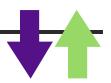
3.11.3.1 Create

Category	Value
Mapping	/Create/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form Object that the Form Element will be associated with;
Request	Form Element Object;
Response	Form Element Object with new Form Element Id;

3.11.4 PUT Mappings

The following mappings are used to update an existing Form Element entry in the database.

3.11.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Form Element Object;
Response	Form Element Object;

3.11.5 DELETE Mappings

The following mappings are used to remove Form Element entries from the database.

3.11.5.1 Remove by Form Element ID

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; The Id of the Form Element Object;
Request	None;
Response	None;

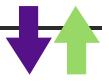## 3.12 Form Element Option

3.12.1 Base Mappings

/FormElementOption

3.12.2 GET Mappings

The following mappings are used to simple get a Form Element Option or List of Form Element Options Objects.

3.12.2.1 Get by Form Element Option Id

Category	Value
Mapping	/ById/FormElementOption/{formElementOptionId}
URL Parameters	formElementOptionId: Integer; Represents the Id of the Form Element Option Object;
Request	None;
Response	Form Element Option Object;

3.12.2.2 Get by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element;
Request	None;
Response	List of Form Element Option Objects

3.12.3 POST Mappings

The following mappings are used to create a new Form Element Option entry into the database.
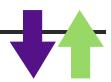
3.12.3.1 Create

Category	Value
Mapping	/Create/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element Object that the Form Element Option will be associated with;
Request	Form Element Option Object;
Response	Form Element Option Object with new Form Element Option Id;

3.12.4 PUT Mappings

The following mappings are used to update an existing Form Element Option entry in the database.

3.12.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Form Element Option Object;
Response	Form Element Option Object;

3.12.5 DELETE Mappings

The following mappings are used to remove Form Element Option entries from the database.

3.12.5.1 Remove by Form Element Option ID

Category	Value
Mapping	/ById/FormElementOption/{formElementOptionId}
URL Parameters	formElementOptionId: Integer; The Id of the Form Element Option Object;
Request	None
Response	None

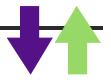## 3.13 Form Flow

3.13.1 Base Mappings

/FormFlow

3.13.2 GET Mappings

The following mappings are used to simple get a Form Flow or List of Form Flow Objects.

3.13.2.1 Get by Form Flow Id

Category	Value
Mapping	/ById/FormFlow/{formFlowId}
URL Parameters	formFlow: Integer; Represents the Id of the Form Flow Object;
Request	None
Response	Form Flow Object;

3.13.2.2 Get by Form Id

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form;
Request	None
Response	List of Form Flow Objects

3.13.3 POST Mappings

The following mappings are used to create a new Form Flow entry into the database.

3.13.3.1 Create

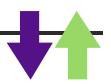| Category	Value
Mapping	/Create/
URL Parameters	None;
Request	Form Flow Object;
Response	Form Flow Object with new Form Flow Id;

3.13.4 PUT Mappings

None;

3.13.5 DELETE Mappings

The following mappings are used to remove Form Flow entries from the database.

3.13.5.1 Remove by Form Flow ID

Category	Value
Mapping	/ById/FormFlow/{formFlowId}
URL Parameters	formFlowId: Integer; The Id of the Form Flow Object;
Request	None
Response	None

3.13.5.2 Remove by Form ID

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; The Id of the Form Object;
Request	None
Response	None

3.13 Question

3.13.1 Base Mappings

/Question

3.13.2 GET Mappings
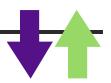
The following mappings are used to simple get a Question or List of Question Objects.

3.13.2.1 Get by Question Id

Category	Value
Mapping	/ById/Question/{questionId}
URL Parameters	questionId: Integer; Represents the Id of the Question Object;
Request	None;
Response	Question Object;

3.13.2.2 Get by Form Id

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form;
Request	None;
Response	List of Question Objects;

3.13.2.3 Get by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer; Represents the Id of the Automation;
Request	None;
Response	List of Question Objects;

3.13.3 POST Mappings

The following mappings are used to create a new Question entry into the database.
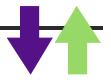
3.13.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None;
Request	Question Object;
Response	Question Object with new Question Id;

3.13.4 PUT Mappings

The following mappings are used to update an existing Question entry in the database.

3.13.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Question Object;
Response	Question Object;

3.13.5 DELETE Mappings

The following mappings are used to remove Question entries from the database.

3.13.5.1 Remove by Question ID

Category	Value
Mapping	/ById/Question/{questionId}
URL Parameters	questionId: Integer; The Id of the Question Object;
Request	None
Response	None

3.13.5.2 Remove by Form ID

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; The Id of the Form Object;
Request	None
Response	None

3.13.5.3 Remove by Automation ID

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer; The Id of the Automation Object;
Request	None
Response	None

3.14 Revision

3.14.1 Base Mappings

/Revision

3.14.2 GET Mappings

The following mappings are used to simple get a Revision or List of Revision Objects.

3.14.2.1 Get by Revision Id

Category	Value
Mapping	/ById/Revision/{revisionId}
URL Parameters	revisionId: Integer; Represents the Id of the Revision;
Request	None;
Response	Revision Objects;

3.14.2.3 Get by Survey Id

Category	Value
Mapping	/ById/Survey/{surveyId}
URL Parameters	surveyId: Integer; Represents the Id of the Survey;
Request	None;
Response	List of Revision Objects

3.14.3 POST Mappings

The following mappings are used to create a new Revision entry into the database.

3.14.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None;
Request	Revision Object;
Response	Revision Object with new Revision Id;

3.14.4 PUT Mappings

The following mappings are used to update an existing Revision entry in the database.

3.14.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Revision Object;
Response	Revision Object;

3.14.5 DELETE Mappings

The following mappings are used to remove Revision entries from the database.

3.14.5.1 Remove by Revision ID

Category	Value
Mapping	/ById/Question/{revisionId}
URL Parameters	revisionId: Integer; The Id of the Revision Object;
Request	None
Response	None

3.14.5.2 Remove by Survey ID

Category	Value
Mapping	/ById/Form/{surveyId}
URL Parameters	surveyId: Integer; The Id of the Survey Object;
Request	None
Response	None