

# **ING Honours project - protocol specification**

David Huistra

June 6, 2017

## **Disclaimer**

This version of the protocol specification now a release candidate!  
Although things might still change, you can use this version to finish  
your implementation.

# Revision History

Revision	Date	Author(s)	Description
0.1	2017/04/24	djhuistra	First draft
0.2	2017/05/02	djhuistra	Camera ready version
0.3	2017/05/09	djhuistra	Two consistency fixes and two new methods. See changelog for more details.
1.0	2017/06/01	djhuistra	Release candidate! Methods returns now consistent. Fixed several examples. See changelog for more details.
1.1	2017/06/06	djhuistra	PIN card and PIN code numbers now Strings in all methods

# Contents

<b>1. Introduction</b>	<b>4</b>
1.1. JSON-RPC Request . . . . .	4
1.2. JSON-RPC Response . . . . .	4
<b>2. Methods</b>	<b>5</b>
2.1. Account Module . . . . .	5
2.1.1. openAccount . . . . .	5
2.1.2. openAdditionalAccount . . . . .	7
2.1.3. closeAccount . . . . .	8
2.2. Access Module . . . . .	9
2.2.1. provideAccess . . . . .	9
2.2.2. revokeAccess . . . . .	10
2.3. Transfer Module . . . . .	11
2.3.1. depositIntoAccount . . . . .	11
2.3.2. payFromAccount . . . . .	12
2.3.3. transferMoney . . . . .	13
2.4. Authentication Module . . . . .	14
2.4.1. getAuthToken . . . . .	14
2.5. Info Module . . . . .	15
2.5.1. getBalance . . . . .	15
2.5.2. getTransactionsOverview . . . . .	16
2.5.3. getUserAccess . . . . .	17
2.5.4. getBankAccountAccess . . . . .	18
<b>A. The Error object</b>	<b>20</b>
A.1. Known Error Codes . . . . .	20
A.2. Examples . . . . .	20
<b>B. Change log</b>	<b>21</b>
B.1. version 0.3 . . . . .	21
B.2. version 1.0 . . . . .	21
B.3. version 1.1 . . . . .	21

# 1. Introduction

This document describes the JSON-RPC 2.0 protocol used in the ING Honours project. It should guide you through the different exposed methods and provide basic pointers to help you to implement a program that communicates using this protocol.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

## 1.1. JSON-RPC Request

In order to execute a remote procedure, you must send an HTTP POST request, containing a json dictionary in compliance to the JSON-RPC 2.0 specification[1]. This implies that each request dictionary must at least contain the following keys:

**jsonrpc** A String specifying the version of the JSON-RPC protocol. This field **MUST** be exactly "2.0" in order to use JSON-RPC 2.0 features.

**method** A String containing the name of the method to be invoked.

**params** A Structured value that holds the parameter values to be used during the invocation of the method. If a method does not require any parameters, an empty array **MUST** be included.

**id** An identifier established by the Client that **MUST** contain a String or Number. Numbers **SHOULD NOT** contain fractional parts.

The 'id' and 'jsonrpc' fields are ommitted in all examples.

## 1.2. JSON-RPC Response

After sending a request, a response object will be returned. The response object consists of a json dictionary containing the following keys:

**jsonrpc** A String specifying the version of the JSON-RPC protocol.

**result** The value of this member is determined by the method invoked on the Server.

**error** This member only exists if an error has occured. The structure of this member is defined in Appendix A. If no error has occured, this member **MAY** also be null.

**id** This member contains the value of the id member in the Request Object. If the id could not be parsed due to a parse error, it will be set to null.

The 'id' and 'jsonrpc' fields are ommitted in all examples.

## 2. Methods

The protocol contains different methods. These methods are all grouped into modules. A module contains the methods involving a specific part of the functionality.

### 2.1. Account Module

#### 2.1.1. openAccount

**Method** openAccount

**Description** Open a bank account. This method also creates a CustomerAccount and PIN card for this account.

**Authentication** (none)

**Parameters** This method accepts the following parameters:

***name*** Customer name

***surname*** Customer surname

***initials*** Customer initials

***dob*** Customer date of birth

***ssn*** Customer SSN

***address*** Customer address

***telephoneNumber*** Customer telephoneNumber

***email*** Customer email

***username*** Customer username

***password*** Customer password

**Returns** A dictionary containing the following members:

***iBAN*** (String) The number of the created bank account

***pinCard*** (String) The number of the created pin card

***pinCode*** (String) The pin code of the created pin card

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.

## Example

```
--> {"method": "openAccount",  
      "params": {  
        "name": "Duck",  
        "surname": "Donald",  
        "initials": "D",  
        "dob": "1954-02-19",  
        "ssn": "571376046",  
        "address": "1313 Webfoot Walk, Duckburg",  
        "telephoneNumber": "+316 12345678",  
        "email": "donald@gmail.com",  
        "username": "duckd",  
        "password": "kwikkwekkwak",  
      }  
}  
<-- {"result": {  
      "iBAN": "NL69INGB0123456789",  
      "pinCard": 1234,  
      "pinCode": 1234,  
    }  
}
```

### 2.1.2. openAdditionalAccount

**Method** openAdditionalAccount

**Description** Open an additional account for an existing customer. Also creates a PIN card for this account.

**Authentication** REQUIRED (account)

**Parameters** This method accepts the following parameters:

**authToken** The authentication token, obtained with getAuthToken

**Returns** A dictionary containing the following members:

**iBAN** (String) The number of the created bank account

**pinCard** (String) The number of the created pin card

**pinCode** (String) The pin code of the created pin card

**Errors** NotAuthorizedError: The authenticated user is not authorized to perform this action.  
**Example**

```
--> {"method":"openAdditionalAccount",
      "params":{"
        "authToken": "some.long.auth.token!%&...",
      }}
<-- {"result": {
      "iBAN": "NL69INGB0123456789",
      "pinCard": "1234",
      "pinCode": "1234",
    }}
```

### 2.1.3. closeAccount

**Method** closeAccount

**Description** Close a bank account. Also invalidates the corresponding pin card. If this is the customers last bank account, it also closes the customer account.

**Authentication** REQUIRED (account)

**Parameters** This method accepts the following parameters:

**authToken** The authentication token, obtained with getAuthToken

**iBAN** (String) The number of the bank account

**Returns** An empty dictionary if successful

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
NotAuthorizedError: The authenticated user is not authorized to perform this action.

**Example**

```
--> {"method":"closeAccount",
      "params":{"
        "authToken": "some.long.auth.token!%&...",
        "iBAN": "NL69INGB0123456789",
      }}
<-- {"result": {}}
```



## 2.2. Access Module

### 2.2.1. provideAccess

**Method** provideAccess

**Description** A customer can share access to one of his bank accounts with another customer. This also creates a corresponding PIN card.

**Authentication** REQUIRED (account)

**Parameters** This method accepts the following parameters:

**authToken** The authentication token, obtained with getAuthToken

**iBAN** (String) The number of the bank account

**username** The username of customer that should get access

**Returns** A dictionary containing the following members:

**pinCard** (String) The number of the created pin card

**pinCode** (String) The pin code of the created pin card

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
NotAuthorizedError: The authenticated user is not authorized to perform this action.  
NoEffectError: The action has no effect. See message. (e.g. access has already been given)

#### Example

```
--> {"method": "provideAccess",
      "params": [
        "authToken": "some.long.auth.token!%&...",
        "iBAN": "NL69INGB0123456789",
        "username": "daisyduck"
      ]}
<-- {"result": {
      "pinCard": "2345",
      "pinCode": "2345",
    }}
```

### 2.2.2. revokeAccess

**Method** revokeAccess

**Description** A customer can revoke his access to a bank account if he is not the primary holder, otherwise the customer can revoke another customers access.

**Authentication** REQUIRED (account)

**Parameters** This method accepts the following parameters:

**authToken** The authentication token, obtained with getAuthToken

**iBAN** (String) The number of the bank account

**username** (optional) The username of customer whoms access should be revoked

**Returns** An empty dictionary if successful

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
NotAuthorizedError: The authenticated user is not authorized to perform this action.  
NoEffectError: The action has no effect. See message. (e.g. username did not have access)

#### Example

```
--> {"method": "revokeAccess",  
      "params": {  
        "authToken": "some.long.auth.token!%&...",  
        "iBAN": "NL69INGB0123456789",  
        "username": "daisyduck"  
      }}  
<-- {"result": {}}
```

## 2.3. Transfer Module

### 2.3.1. depositIntoAccount

**Method** depositIntoAccount

**Description** Deposit money into a bank account.

**Authentication** REQUIRED (pin)

**Parameters** This method accepts the following parameters:

**iBAN** The number of the bank account

**pinCard** (String) The number of the pin card

**pinCode** (String) The code of the pin card

**amount** The amount that is deposited.

**Returns** An empty dictionary if successful

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
InvalidPINError: An invalid PINcard, -code or -combination was used.

**Example**

```
--> {"method":"depositIntoAccount",
      "params":{"
        "iBAN": "NL69INGB0123456789",
        "pinCard": "1234",
        "pinCode": "1234",
        "amount": 313,00
      }}
<-- {"result": {}}
```

### 2.3.2. payFromAccount

**Method** payFromAccount

**Description** Pay from bank account

**Authentication** REQUIRED (pin)

**Parameters** This method accepts the following parameters:

**sourceIBAN** (String) The number of the bank account

**targetIBAN** (String) The number of the target bank account

**pinCard** (String) The number of the pin card

**pinCode** (String) The code of the pin card

**amount** The amount that is deposited.

**Returns** An empty dictionary if successful

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
InvalidPINError: An invalid PINcard, -code or -combination was used.

#### Example

```
--> {"method": "payFromAccount",  
      "params": {  
        "sourceIBAN": "NL69INGB0123456789",  
        "targetIBAN": "NL98INGB1234567890",  
        "pinCard": "1234",  
        "pinCode": "1234",  
        "amount": 313,00  
      }}  
<-- {"result": {}}
```

### 2.3.3. transferMoney

**Method** transferMoney

**Description** Transfer money between two bank accounts.

**Authentication** REQUIRED (account)

**Parameters** This method accepts the following parameters:

**authToken** The authentication token, obtained with getAuthToken

**sourceIBAN** The number of the bank account

**targetIBAN** The number of the target bank account

**targetName** The name of the target bank account holder

**amount** The amount that is deposited.

**description** Description of the transaction.

**Returns** An empty dictionary if successful

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
NotAuthorizedError: The authenticated user is not authorized to perform this action.

**Example** (per execution: with jsonrpc and id tag).

```
--> {"jsonrpc": "2.0", "method": "transferMoney", "id": 1,
    "params": {
        "authToken": "some.long.auth.token!%&...",
        "sourceIBAN": "NL69INGB0123456789",
        "targetIBAN": "NL98INGB1234567890",
        "targetName": "Donald D. Duck",
        "amount": 313,00,
        "description": "Some description"
    }}
<-- {"jsonrpc": "2.0", "result": {}, "id": 1}
```

## 2.4. Authentication Module

### 2.4.1. getAuthToken

**Method** getAuthToken

**Description** Attempts to log in and returns an authentication token.

**Authentication** (none)

**Parameters** This method accepts the following parameters:

***username*** The username of the customer

***password*** The password of the customer

**Returns** A dictionary containing the following members:

***authToken*** The authToken string

**Errors** AuthenticationError: The user could not be authenticated. Invalid username, password or combination.

#### Example

```
--> {"method": "getAuthToken",  
      "params": {  
        "username": "duckd",  
        "password": "kwikkwekkwak",  
      }}  
<-- {"result": {"authToken": "some.long.auth.token!%&..."}}
```

## 2.5. Info Module

### 2.5.1. getBalance

**Method** getBalance

**Description** Retrieves the balance of a bank account.

**Authentication** REQUIRED (account)

**Parameters** This method accepts the following parameters:

**authToken** The authentication token, obtained with getAuthToken

**iBAN** The number of the bank account

**Returns** A dictionary containing the following members:

**balance** The balance of the account

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
NotAuthorizedError: The authenticated user is not authorized to perform this action.

#### Example

```
--> {"method": "getBalance",  
    "params": {  
        "authToken": "some.long.auth.token!%&...",  
        "iBAN": "NL69INGB0123456789",  
    }}  
<-- {"result": {"balance": -2000,00}}
```

## 2.5.2. getTransactionsOverview

**Method** getTransactionsOverview

**Description** Retrieves overview of recent transactions of a bank account.

**Authentication** REQUIRED (account)

**Parameters** This method accepts the following parameters:

**authToken** The authentication token, obtained with getAuthToken

**iBAN** The number of the bank account

**nrOfTransactions** The number of transactions requested

**Returns** An array of dictionaries containing the following members:

**sourceIBAN** The number of the source bank account

**targetIBAN** The number of the target bank account

**targetName** The name of the target bank account holder

**date** The date of the transaction

**amount** The amount transferred

**description** Description of the transaction

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
NotAuthorizedError: The authenticated user is not authorized to perform this action.

### Example

```
--> {"method": "getTransactionsOverview",
      "params": {
        "authToken": "some.long.auth.token!%&...",
        "iBAN": "NL69INGB0123456789",
        "nrOfTransactions": 1,
      }}
<-- {"result": [{
  "sourceIBAN": "NL69INGB0123456789",
  "targetIBAN": "NL98INGB1234567890",
  "targetName": "Donald D. Duck",
  "date": "2014-07-02T018:00:00+02:00",
  "amount": 5.00,
  "description": "Salary"
}, {...}, {...}]}
```



### 2.5.3. getUserAccess

**Method** getUserAccess

**Description** Retrieves an overview of all bank accounts a user has access to. User can only request this for himself.

**Authentication** REQUIRED (account)

**Parameters** This method accepts the following parameters:

**authToken** The authentication token, obtained with getAuthToken

**Returns** An array of dictionaries containing the following members:

**iBAN** The IBAN of the bank account.

**owner** The username of the accounts owner.

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
NotAuthorizedError: The authenticated user is not authorized to perform this action.

#### Example

```
--> {"method": "getUserAccess",  
      "params": {  
        "authToken": "some.long.auth.token!%&..."  
      }}  
<-- {"result": [{  
      "iBAN": "NL69INGB0123456789",  
      "owner": "duckd"  
    }, {...}, {...}]}
```

### 2.5.4. getBankAccountAccess

**Method** getBankAccountAccess

**Description** Retrieves an overview of all users who have access to a bank account. Only the owner of an account can perform this request.

**Authentication** REQUIRED (account)

**Parameters** This method accepts the following parameters:

**authToken** The authentication token, obtained with getAuthToken

**iBAN** The IBAN of the bank account.

**Returns** An array of dictionaries containing the following members:

**username** The username of a UserAccount that has access to this bank account.

**Errors** InvalidParamValueError: One or more parameter has an invalid value. See message.  
NotAuthorizedError: The authenticated user is not authorized to perform this action.

**Example**

```
--> {"method": "getBankAccountAccess",
      "params": {
        "authToken": "some.long.auth.token!%&...",
        "iBAN": "NL69INGB0123456789",
      }}
<-- {"result": [{
      "username": "duckd"
    }, {...}, {...}]}
```

# Bibliography

- [1] JSON-RPC Working Group (2010), JSON-RPC 2.0 specification.  
Available at <http://www.jsonrpc.org/specification>

## A. The Error object

An Error object has the following fields[1]:

**code** A Number that indicates the error type that occurred. This MUST be an integer.

**message** A String providing a short description of the error. The message SHOULD be limited to a concise single sentence.

**data** A value that contains additional information about the error. The value of this member is defined by the Server, this member MAY be omitted.

### A.1. Known Error Codes

Error Code	Name	Description
418	InvalidParamValueError	One or more parameter has an invalid value. See message.
419	NotAuthorizedError	The authenticated user is not authorized to perform this action.
420	NoEffectError	The action has no effect. See message.
421	InvalidPINError	An invalid PINcard, -code or -combination was used.
422	AuthenticationError	The user could not be authenticated. Invalid username, password or combination.
500	OtherError	An unexpected error occurred, see error details.
-32700	ParseError	An error occurred while parsing the JSON input.
-32600	InvalidRequestError	The received JSON is not a valid JSON-RPC Request.
-32601	MethodNotFoundError	The requested remote-procedure does not exist.
-32602	InvalidParamsError	Invalid method parameters.
-32603	ServerError	Internal JSON-RPC error.

Table A.1.: Overview of different errors.

### A.2. Examples

```
--> { "method": "foobar" }
<-- { "error": { "code": -32601, "message": "Method not found" } }
```

```
--> { "method": "getBalance", "params": { "authToken": "some.long.auth.token
!%&...", "iBAN": "NL00INGB0123456789" } }
<-- { "error": { "code": 418, "message": "One or more parameter has an invalid
value.", "data": "IBAN does not exist" } }
```

## B. Change log

An overview of changes per version

### B.1. version 0.3

- **2.1.2** openAdditionalAccount now has NotAuthorizedError
- **2.1.3** closeAccount paramter name change: bankAccountNumber to iBAN
- **2.5** Info module: Two additional info methods added. getUserAccess and getBankAccountAccess
- **2.3.2** Example fixed. return parameter amount was capitalized.

### B.2. version 1.0

- **All methods** All methods now return an (empty) dictionary or an array of dictionaries instead of a single value
- **2.5.3** getUserAccess no longer requires username parameter.
- **2.1.1** Example fixed. Several parameter values did not have quotation marks
- **2.3.1** Example fixed. return parameter amount was capitalized.
- **2.3.2** Example fixed. return parameter amount was capitalized.

### B.3. version 1.1

- **All methods** pinCode and PinCard numbers changed to String
- **2.3.2** Example (finally) fixed. return parameter amount was capitalized.