

# Handleiding NOOB zonder encryptie

Naïeve Ontzettend Onveilige Bank

<b>H1 Socket.io</b>	<b>2</b>
<b>H2 Verbinden</b>	<b>3</b>
<b>H3 Requests</b>	<b>4</b>
§3.1 Withdraw	4
§3.2 Get balance	5
<b>H4 Responses</b>	<b>6</b>
§4.1 Withdraw	6
§4.2 Get balance	7
<b>H5 Walkthrough code</b>	<b>8</b>
<b>H6 Support</b>	<b>11</b>
<b>H7 Changelog</b>	<b>12</b>

# H1 Socket.io

De verbinding met de NOOB gaat via socket.io. Socket.io is een abstractie van websockets, alleen heeft socket.io iets meer opties en veiligheden. De volgende libraries zijn aangeraden:

- Javascript: <https://github.com/socketio/socket.io-client>
- Java: <https://github.com/socketio/socket.io-client-java>
- C++: <https://github.com/socketio/socket.io-client-cpp>
- Python: <https://python-socketio.readthedocs.io/en/latest/client.html>

Bij het ontwerpen van de NOOB is er vanuit gegaan dat er één verbinding per land wordt opgezet. Omdat elke klas de implementatie van deze verbinding zelf mag invullen, zal er in dit document verder gerefereerd worden als een 'node'.

Per land zal er één verbinding worden toegestaan, er moet dus een node worden gemaakt zodat iedereen kan communiceren met de andere landen. Met deze node zal de communicatie binnen een land worden afgehandeld en de communicatie via de NOOB naar andere landen worden gestuurd.

## H2 Verbinden

Elke taal heeft net weer een andere manier van verbinden kijk dus goed in de readme van de bovenstaande libraries.

# H3 Requests

Om de communicatie tussen de landen goed te laten verlopen is er een protocol opgesteld, dit protocol gebruikt JSON objects. Om een request te sturen moet er een emit request. Een emit request bestaat altijd uit een event naam en de data (Bijvoorbeeld:

`socket.emit("withdraw", data)`). Om van de NOOB gebruik te maken zal eerst de landnaam en het IP van de landnode moeten worden doorgegeven aan de docenten, hierna kunnen er pas `withdraws` en andere requests worden gemaakt.

## §3.1 Withdraw

Om een `withdraw` emit te sturen zal als event "`withdraw`" worden ingevoerd en de volgende data als JSON object worden meegestuurd:

```
{
  'body': {
    'pin': String,
    'account': String,
    'amount': Float
  },
  'header': {
    'originCountry': String,
    'originBank': String,
    'receiveCountry': String,
    'receiveBank': String
  }
}
```

Voorbeeld:

```
{
  'body': {
    'pin': '0000',
    'account': '01234567',
    'amount': 13.37
  },
  'header': {
    'originCountry': 'NL',
    'originBank': 'INGB',
    'receiveCountry': 'DE',
    'receiveBank': 'DEBA'
  }
}
```

## §3.2 Get balance

Om een get balance emit te sturen zal als event “balance” worden ingevoerd en de volgende data als JSON object worden meegestuurd:

```
{
  'body': {
    'pin': String,
    'account': String
  },
  'header': {
    'originCountry': String,
    'originBank': String,
    'receiveCountry': String,
    'receiveBank': String
  }
}
```

Voorbeeld:

```
{
  'body': {
    'pin': '0000',
    'account': '01234567'
  },
  'header': {
    'originCountry': 'NL',
    'originBank': 'INGB',
    'receiveCountry': 'DE',
    'receiveBank': 'DEBA'
  }
}
```

# H4 Responses

## §4.1 Withdraw

Responses op 'withdraw' events hebben het volgende format:

```
{
  'body': {
    'code': Int,
    'message': String
  },
  'header': {
    'originCountry': String,
    'originBank': String,
    'receiveCountry': String,
    'receiveBank': String,
    'action': String
  }
}
```

Voorbeeld:

```
{
  'body': {
    'code': 200,
    'message': 'Success'
  },
  'header': {
    'originCountry': 'NL',
    'originBank': 'INGB',
    'receiveCountry': 'DE',
    'receiveBank': 'DEBA',
    'action': 'withdraw'
  }
}
```

## §4.2 Get balance

Responses op 'balance' events hebben het volgende format:

```
{
  'body': {
    'code': String,
    'message': String
    'balance': Float
  },
  'header': {
    'originCountry': String,
    'originBank': String,
    'receiveCountry': String,
    'receiveBank': String,
    'action': String
  }
}
```

Voorbeeld:

```
{
  'body': {
    'code': 200,
    'message': 'Success',
    'balance': 999.99
  },
  'header': {
    'originCountry': 'NL',
    'originBank': 'INGB',
    'receiveCountry': 'DE',
    'receiveBank': 'DEBA',
    'action': 'balance'
  }
}
```

Let op de correcte afhandeling van de mogelijke afwezigheid van de 'balance' key/value bij andere statuscodes dan 200.



## H5 Walkthrough code

In de Github repository is een voorbeeld Python script te vinden dat withdraw requests uitvoert. Hieronder volgt een korte beschrijving over de werking van de code, hoe het opgezet moet worden en hoe je zelf requests naar de noob kan sturen.

Voor dit programma moet je de python-socketio library hebben. Installeer dit door het volgende commando: `pip3 install "python-socketio[client]"`.

In dit script wordt als eerst een connectie met de NOOB opgezet door middel van `io.connect`. Hierbij moet de url aangepast met het ip van de server waar de NOOB op draait. In dit script wordt zowel een withdraw request als response naar de NOOB gestuurd. Hieronder is de body van het verstuurd frame:

```
withdrawBody = {
    'header': {
        'originCountry': 'NL',
        'originBank': 'INGB',
        'receiveCountry': 'DE',
        'receiveBank': 'DEBA'
    },
    'body': {
        'pin': '1234',
        'account': '123456',
        'amount': 123.45
    }
}
```

Dit wordt verstuurd met: `io.emit('withdraw', withdrawBody)`. Als je bijvoorbeeld de balance bij een ander land wilt ophalen zou je de emit functie kunnen veranderen met het volgende: `io.emit('balance', balanceBody)`. Hierbij moet de body veranderd worden zoals hiervoor in hoofdstuk 3 is beschreven.

Indien land A een frame ontvangt van land B, bijvoorbeeld een withdraw frame, dan zal land A ook een response naar land B terug moeten sturen. De land moet dit altijd terugsturen over de response action. In de onderstaande code wordt door land A een response naar land B gestuurd. De response action moet bestaan uit een status code en een message. De message mag vrij worden ingevuld maar moet wel een beschrijving van het probleem (foute pin, account bestaat niet, niet genoeg balance, etc) of een bevestiging (balance is: XXX, withdraw gelukt) geven. De status code moet een HTTP status code zijn ([https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)), deze codes zijn makkelijk te gebruiken en begrijpen.

```

response = {
  'header': {
    'originCountry': '$CTRY',
    'originBank': '$BANK',
    'receiveCountry': data['header']['originCountry'],
    'receiveBank': data['header']['originBank'],
    'action': 'withdraw'
  },
  'body': {
    'code': 200,
    'message': 'Success'
  }
}
io.emit('response', response)

```

Een voorbeeld van een getbalance response en request.

Land A:

```

getBalanceBody = {
  "body": {
    "pin": "1234",
    "account": "123456"
  },
  "header": {
    "originCountry": "DE",
    "originBank": "DEBA",
    "receiveCountry": "NL",
    "receiveBank": "INGB"
  }
}
io.emit("balance", getBalanceBody)

```

Land B stuurt terug met:

```

{
  "body": {
    "code": 200,
    "message": "Success",
    "balance": 9999.99
  },
  "header": {
    "originCountry": "NL",
    "originBank": "INGB",
    "receiveCountry": "DE",
    "receiveBank": "DEBA",
    "action": "balance"
  }
}

```

Of bijvoorbeeld

```
{
  "body": {
    "code": 404,
    "message": "Account bestaat niet"
  },
  "header": {
    "originCountry": "DE",
    "originBank": "DEBA",
    "receiveCountry": "NL",
    "receiveBank": "ABNA",
    "action": "balance"
  }
}
```

## H6 Support

Kom je iets tegen wat niet werkt? Of lukt het niet op socket.io op te zetten? Maak een issue aan op de github: <https://github.com/luukk/noob/issues/new>.

Voor het registreren van een (test)land-node, neem contact op met je docent.

# H7 Changelog

## 20-03-2020:

- Eerste versie

## 23-03-2020:

- **Toegevoegd:** Bij elke request en response wordt nu ook een `originCountry` en `originBank` meegegeven om responses te kunnen verwerken.
- **Aangepast:** Leesbaarheid handleiding verbeterd; code in monospace