IOU Central

API v2.0 – Documentation

ABOUT THIS DOCUMENT

The purpose of this document is to provide a reference for the IOU Central API library.

DOCUMENT HISTORY

Version	Date Modified	Author/Editor	Description
1.0	15/Jan/2014	Sam Kawtharani	• Initial v2.0 documentation based on 1.1. Added support for multiple personal guarantors and repayment information

TABLE OF CONTENTS

SETUP

Every API command expects a security token in the request parameters for authenticating brokers against our platform.

This means, that any external system will need to create an additional profile field that stores the token for every user and pass it as part of the API calls.

DATA FORMATS

All requests are by default, provided as JSON POST.

AUTHENTICATION

All requests to IOU Central service require access tokens.

The token should be passed as part of the data parameters in the JSON POST request.

Example:

```
{
        token: “ae6a724ab267ed555c03a4fc2fdf3b8d86e4bd0be1ca0c6edc00464d722aa123”,
        …
        …
}
```

In case of invalid user tokens, the API will return the following error response:

Code: 401 Unauthorized

Content: {"error":"Token is invalid."}

DEVELOPMENT SANDBOX - ADDITIONAL AUTHENTICATION

Access to the development sandbox requires an htaccess authentication header in the URL.

Example: USERNAME:PASSWORD@demo.ioucentral.com

Python Example:

```
requests.post('API_URL', data={'token':'SECURITY_TOKEN'}, auth=HTTPBasicAuth('USERNAME', 'PASSWORD'))
```

API COMMANDS

LOAN CALCULATOR

Title	Loan Calculator
Description	The following API command is a Loan Calculator that returns a preliminary loan product based on loan amount and term. On success the command will return a preliminary IOU Loan Product.
URL	/api/calculators/loans
Method	GET
URL Parameters	**Required:** token=[string]

Data Parameters	• **amount:** *Numeric, The loan amount.*
	• **term:** *Numeric, The term of loan. 6 or 12 months.*
	
	<u>Example:</u>
	`{`
	`  "loan":{`
	`        "amount":25000,`
	`        "term":12`
	`  }`
	`}`

Success Response	**Code:** 200 OK
	
	Content:
	`{`
	`  "loan_values": {`
	`        "amount": 25000,`
	`        "closing_fee": 1237.5,`
	`        "daily_gfee": 7.94,`
	`        "daily_gfee_2": 11.91,`
	`        "daily_prin_int": 106.86,`
	`        "pay_count": 252,`
	`        "rate": 14.99,`
	`        "term": 12`
	`  }`
	`}`
	closing_fee = Loan Origination fee.

	daily_gfee = Loan Guaranty Fee – Lower Limit **daily_gfee2** = Loan Guaranty Fee – Upper Limit **daily_prin_int** = The daily loan principal including interest and excluding Guaranty Fee. (Clients can use the lower and upper Guaranty Fee values to show the total payment range). **pay_count** = Total number of Loan Payments. **rate** = The loan's interest rate.

Error Response	**Code:** 412 Precondition Failed
	
	Content:
	
	`{`
	`  "errors": {`
	`        "loan": ["Field 1 Error Message", "Field 2 Error Message" , ...],`
	`  }`
	`}`

Sample Call	`$.ajax({`
	` url: "/api/calculators/loans",`
	` dataType: "json",`
	` data : {`
	`  "token":`
	`  "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9`
	`   e756c9525c0f44d3aaa7",`
	
	`  "loan":{`
	`      "amount":25000,`
	`      "term":12`
	`  }`
	` },`
	` type : "GET",`
	` success : function(r) {`
	`  console.log(r);`
	` }`
	`});`

CREATE USER

Title	Step 1 - Create User Account
Description	The following API command creates the user/company account that will be associated with the loan request. A user account is represented by a company, user, owner(s), company address,

	and owner/user address. On success the command will return the IOU company/user id.
URL	/api/users
Method	POST
URL Parameters	N/A
Data Parameters	<u>**Company:**</u> • **first_name**: *String, Company's Legal Name.* • **last_name:** *String, Company's DBA Name.* • **email:** *Alphanumeric, User's email.* • **owner_since:** *String, Ownership date in the following format: MM/DD/YYYY.* • **industry:** *A value from the list below:* 　○ retail_store 　○ restaurant_or_cafe 　○ medical_office 　○ dental_office 　○ online_service 　○ fast_food 　○ fine_dining 　○ clothing_store_or_boutique 　○ furniture_store 　○ grocery_store 　○ auto_repair_or_parts_shop 　○ spa 　○ hair_or_nail_salon 　○ day_care_center 　○ fun_entertainment 　○ hotel 　○ gas_station 　○ liqour_store 　○ chiropractor • **type_inc:** *A value from the list below:* 　○ inc_type_corporation 　○ inc_type_llc 　○ inc_type_general_partnership 　○ inc_type_llp 　○ inc_type_sole_proprietorship 　○ inc_type_other

- **birthdate:** *String, Company's inception date in the following format: MM/DD/YYYY.*
- **phone_mobile:** *Number, Company's 10 digits cell phone number.*
- **phone_work:** *Number, Company's 10 digits work phone number. An optional extension number (1-5 digits) can be appended at the end.*
- **taxid:** *Number, A unique 9 digits tax id number.*
- **state_inc:** *String, Company's sate of incorporation. 2 letters state abbreviation.*
- **description:** *String, A short description about the company and field of work.*
- **monthly_cc_volume:** *Number, The company's monthly credit card volum.*
- **average_ticket_item:** *Number, The company's average sale ticket items.*
- **gross_revenue:** *Hash, A hash containing the previous and current gross annual revenues (estimated for current year).*

Company Address:

- **civic:** *Numeric, Address civic number.*
- **street:** *String, Address street name.*
- **unit:** *String, Address unit/suite/apartment number.*
- **city:** *String, Address city name.*
- **zip:** *Numeric, Address 5 digits zip code.*
- **province:** *String, The 2 letters state abbreviation.*
- **country:** *String, Default to US.*
- **contact_name:** *String, Contact's full name. (Send null if empty)*
- **valid_from:** *String, At address since date in the following format: MM/DD/YYYY.*

User:

The user param takes an array/list of owners. Each owner has the following parameters:

- **first_name:** *String, Owner's First Name.*
- **last_name:** *String, Owner's Last Name.*
- **phone_mobile:** *Number, Owner's 10 digits mobile phone number.*
- **phone:** *Number, Owner's 10 digits home phone number.*
- **phone_work:** *Number, Owner's 10 digits work phone number. . An optional extension number (1-5 digits) can be appended at the end.*
- **birthdate:** *String, Owner's birthdate in the following format: MM/DD/YYYY.*
- **ssn:** *Number, A unique 9 digits Social Security Number.*
- **gender:** *String, Accepted Values: M for Male, F for Female*

- **owner_percent:** *Number, An owner ship percentage of the business.*
- **primary_guarantor:** *Boolean(1 or 0), A flag to designate if the owner is the primary loan guarantor.*
- **address:** *Hash, A hash containing the address data for the corresponding owner:*
    - **civic:** *Numeric, Address civic number.*
    - **street:** *String, Address street name.*
    - **unit:** *String, Address unit/suite/apartment number.*
    - **city:** *String, Address city name.*
    - **zip:** *Numeric, Address 5 digits zip code.*
    - **province:** *String, The 2 letters state abbreviation.*
    - **country:** *String, Default to US.*
    - **contact_name:** *String, Contact's full name. (Send null if empty)*
        - **valid_from:** *String, At address since date in the following format: MM/DD/YYYY.*

<u>Example:</u>

```
{
   "company":{
         "first_name":"Company Name",
         "last_name":"DBA Name",
         "email":"curtis@sipesfadel.biz",
         "owner_since":"02/23/2000",
         "industry":"chiropractor",
         "type_inc":"inc_type_llc",
         "birthdate":"02/24/2009",
         "phone_mobile":"6519674715",
         "phone_work":"6519674715",
         "taxid":"123456789",
         "state_inc":"NY",
         "description":"Italian cuisine restaurant."
         "monthly_cc_volume":"150000",
         "average_ticket_item":"1000",
         "gross_revenue":{
                 "2013":700000,
                 "2014":642000
         }
   },

   "company_address":{
         "civic":"662",
         "street":"70451 Jaden Row",
         "unit":"482",
         "city":"Ebertburgh",
         "zip":"94615",
         "province":"IL",
        "country":"us",
        "contact_name":null,
        "valid_from":"05/16/2003"
    },

    "user":[
        {
            "first_name":"Russ",
            "last_name":"Anderson",
            "phone_mobile":"4436595043",
            "phone":"1234323234",
            "phone_work":"34324323242",
            "birthdate":"09/30/1983",
            "ssn":"123456987",
            "gender":"M",
            "owner_percent":"79",
            "primary_guarantor ":1
            "address":{
               "civic":"662",
               "street":"70451 Jaden Row",
               "unit":"482",
               "city":"Ebertburgh",
               "zip":"94615",
               "province":"IL",
               "country":"us",
               "contact_name":"Russ Anderson",
               "valid_from":"05/16/2003"
            }
        },
        {
            "first_name":"Rachel",
            "last_name":"Anderson",
            "phone_mobile":"4436595077",
            "phone":"1234323239",
            "phone_work":"34324323282",
            "birthdate":"09/30/1982",
            "ssn":"123489987",
            "gender":"F",
            "owner_percent":"21",
            "primary_guarantor ":0
            "address":{
               "civic":"662",
               "street":"70451 Jaden Row",
               "unit":"482",
               "city":"Ebertburgh",
               "zip":"94615",
               "province":"IL",
               "country":"us",
               "contact_name":"Rachel Anderson",
                "valid_from":"05/16/2003"
             }
          }
   ]
}
```

Success Response

Code: 200 OK

Content: {"company_id": "cmpDA8226579"}

Error Response

Code: 412 Precondition Failed

Content:

```
{
   "errors": {
         "company": [“Field 1 Error Message”, “Field 2 Error Message”, …],

         "company_address": [“Field 1 Error Message”, “Field 2 Error Message”,
…],

         "user": [“Field 1 Error Message”, “Field 2 Error Message”, …],

         "user_address": [“Field 1 Error Message”, “Field 2 Error Message”, …]
   }
}
```

Sample Call

```
$.ajax({
 url: "/api/users",
 dataType: "json",
 data : {
  "token":
  "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9
   e756c9525c0f44d3aaa7",

  "company":{
        "first_name":"Company Name",
        "last_name":"DBA Name",
        "email":"curtis@sipesfadel.biz",
        "owner_since":"02/23/2000",
        "industry":"chiropractor",
        "type_inc":"inc_type_llc",
        "birthdate":"02/24/2009",
        "phone_mobile":"6519674715",
        "phone_work":"6519674715",
        "taxid":"123456789",
        "state_inc":"NY",
        "description":"Italian cuisine restaurant."
        "monthly_cc_volume":"150000",
        "average_ticket_item":"1000",
        "gross_revenue":{
                "2013":700000,
                "2014":642000
        }
},

"company_address":{
        "civic":"662",
        "street":"70451 Jaden Row",
        "unit":"482",
        "city":"Ebertburgh",
        "zip":"94615",
        "province":"IL",
        "country":"us",
        "contact_name":null,
        "valid_from":"05/16/2003"
},

"user":[
        {
            "first_name":"Russ",
            "last_name":"Anderson",
            "phone_mobile":"4436595043",
            "phone":"1234323234",
            "phone_work":"34324323242",
            "birthdate":"09/30/1983",
            "ssn":"123456987",
            "gender":"M",
            "owner_percent":"79",
            "primary_guarantor ":1
            "address":{
                "civic":"662",
                "street":"70451 Jaden Row",
                "unit":"482",
                "city":"Ebertburgh",
                "zip":"94615",
                "province":"IL",
                "country":"us",
                "contact_name":"Russ Anderson",
                "valid_from":"05/16/2003"
            }
        },
        {
            "first_name":"Rachel",
            "last_name":"Anderson",
            "phone_mobile":"4436595077",
            "phone":"1234323239",
            "phone_work":"34324323282",
```

	```
	          "birthdate":"09/30/1982",
	          "ssn":"123489987",
	          "gender":"F",
	          "owner_percent":"21",
	          "primary_guarantor ":0
	          "address":{
	             "civic":"662",
	             "street":"70451 Jaden Row",
	             "unit":"482",
	             "city":"Ebertburgh",
	             "zip":"94615",
	             "province":"IL",
	             "country":"us",
	             "contact_name":"Rachel Anderson",
	             "valid_from":"05/16/2003"
	          }
	      }
	  ]
	 },
	 type : "POST",
	 success : function(r) {
	  console.log(r);
	 }
	});
	```

GET USER

Title	Get User Information
Description	The following API fetches the user/company account information created by the Create User API command. On success the command will return a JSON representation of the user object.
URL	/api/users/**USER_ID**
Method	GET
URL Parameters	**Required:** user_id=[string] example: cmpDA8226579
Data Parameters	N/A

Success Response

Code: 200 OK

Content:

```
{
  "company":{
      "first_name":"Company Name",
      "last_name":"DBA Name",
      "email":"curtis@sipesfadel.biz",
      "owner_since":"02/23/2000",
      "industry":"chiropractor",
      "type_inc":"inc_type_llc",
      "birthdate":"02/24/2009",
      "phone_mobile":"6519674715",
      "phone_work":"6519674715",
      "taxid":"123456789",
      "state_inc":"NY",
      "description":"Italian cuisine restaurant."
      "monthly_cc_volume":"150000",
      "average_ticket_item":"1000",
      "gross_revenue":{
            "2013":700000,
            "2014":642000
      }
  },

  "company_address":{
      "civic":"662",
      "street":"70451 Jaden Row",
      "unit":"482",
      "city":"Ebertburgh",
      "zip":"94615",
      "province":"IL",
      "country":"us",
      "contact_name":null,
      "valid_from":"05/16/2003"
  },

  "user":[
      {
          "first_name":"Russ",
          "last_name":"Anderson",
          "phone_mobile":"4436595043",
          "phone":"1234323234",
          "phone_work":"34324323242",
          "birthdate":"09/30/1983",
          "ssn":"123456987",
          "gender":"M",
          "owner_percent":"79",
```

	```
	            "primary_guarantor ":1
	            "address":{
	               "civic":"662",
	               "street":"70451 Jaden Row",
	               "unit":"482",
	               "city":"Ebertburgh",
	               "zip":"94615",
	               "province":"IL",
	               "country":"us",
	               "contact_name":"Russ Anderson",
	               "valid_from":"05/16/2003"
	            }
	        },
	        {
	            "first_name":"Rachel",
	            "last_name":"Anderson",
	            "phone_mobile":"4436595077",
	            "phone":"1234323239",
	            "phone_work":"34324323282",
	            "birthdate":"09/30/1982",
	            "ssn":"123489987",
	            "gender":"F",
	            "owner_percent":"21",
	            "primary_guarantor ":0
	            "address":{
	               "civic":"662",
	               "street":"70451 Jaden Row",
	               "unit":"482",
	               "city":"Ebertburgh",
	               "zip":"94615",
	               "province":"IL",
	               "country":"us",
	               "contact_name":"Rachel Anderson",
	               "valid_from":"05/16/2003"
	            }
	        }
	   ]
	}
	```

Error Response	**Code:** 404 Not Found
	
	Content:
	
	{
	  "errors":"User **USER_ID** Not Found for broker **BROKER_EMAIL.**"
	}

Sample Call	
	```
	$.ajax({
	  url: "/api/users/cmpDA8226579",
	```

```
 dataType: "json",
 data : {
  "token":
  "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9
   e756c9525c0f44d3aaa7",
 },
 type : "GET",
 success : function(r) {
  console.log(r);
 }
});
```

CREATE LOAN REQUEST

Title	Step 2A - Create Loan Request
Description	The following API command creates a loan request for the user in Step 1. On success the command will return the IOU Loan ID.
URL	/api/users/**COMPANY_ID**/loans
Method	POST
URL Parameters	**Required:** company_id=[string] example: cmpDA8226579

Data Parameters	• **amount:** *Numeric, The requested loan amount.*
	• **length_max:** *Numeric, The maximum term of the requested loan. 6 or 12 months.*
	• **description:** *Numeric, A short description about loan reasons.*
	
	Example:
	
	`{`
	`  "loan":{`
	`        "amount":50000,`
	`        "length_max":6,`
	`        "description":"Some Info Why"`
	`  }`
	`}`

Success Response	**Code:** 200 OK

	Content: {"loan_id": "cmpda8226579"}

Error Response	**Code:** 412 Precondition Failed
	
	Content:
	
	`{`
	`  "errors": {`
	`       "loan": [“Field 1 Error Message”, “Field 2 Error Message” , ...],`
	`  }`
	`}`

Sample Call	see below

Sample Call

```
$.ajax({
 url: "/api/users/cmpDA8226579/loans",
 dataType: "json",
 data : {
  "token":
  "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9
   e756c9525c0f44d3aaa7",

  "loan":{
     "amount":50000,
     "length_max":6,
     "description":"some info"
  }
 },
 type : "POST",
 success : function(r) {
  console.log(r);
 }
});
```

GET LOAN REQUEST

Title	Get Loan Request Information and Status
Description	The following API fetches the loan information created by the Create Loan Request API command. On success the command will return a JSON representation of the loan object along with its status. If the loan is in the funded state, a borrower invitation URL will be included in the response. If the loan funds were accepted, the repayment information will be included in

	the response. <u>Application Statuses:</u> 1. **api_added:** Loan has been created by the API. 2. **activated:** Soft Prequal was accepted and the application is under review. 3. **conditional_approval**: Loan has been conditionally approved and is pending final underwriting approval. 4. **rejected:** Application has been rejected by IOU team. 5. **open:** Loan has been approved by underwriters and is pending funding. 6. **funded:** Loan has been funded by IOU and is pending borrower's acceptance. 7. **closed:** Funded loan has been accepted by borrower. (*Agreements are available at this stage*) 8. **collecting:** Loan has been funded and repayment schedule initiated. <u>Repayment Information:</u> 1. **cd_gfee_payment:** The daily guaranty fee payment 2. **cd_last_pay_amount:** Last loan payment amount 3. **cd_pay_date_first:** First loan payment date 4. **cd_pay_date_last:** Last loan payment date 5. **cd_payment_count:** Number of loan payments 6. **cd_payment_with_gfee:** Total daily loan payment including guaranty fee 7. **cd_regular_repayment_amount:** Daily loan principal payment (exluding guaranty fee) 8. **cd_total_interest:** Total loan interest 9. **total_outstanding_balance:** Loan outstanding balance
URL	/api/loans/**LOAN_ID**
Method	GET
URL Parameters	**Required:** loan_id=[string] example: cmpda8226579
Data Parameters	N/A

Success Response	**Code:** 200 OK
	
	Content:
	
	`{`
	`  "loan":{`

```
        "status":"conditional_approval"
         "amount":50000,
        "length_max":6,
        "description":"Some Info Why",
        "amount_approved":50000,
        "invitation_url":"ioucentral.com/emails/ACTIVATION_CODE/confirm",
        "repayment_info": {
                "cd_gfee_payment": 8.75,
                "cd_last_pay_amount": 22.47,
                "cd_pay_date_first": "01/21/2014",
                "cd_pay_date_last": "07/14/2014",
                "cd_payment_count": 122,
                "cd_payment_with_gfee": 109.14,
                "cd_regular_repayment_amount": 100.39,
                "cd_total_interest": 361.69,
                "total_outstanding_balance": 11800.0
      },
   }
}
```

Error Response

Code: 404 Not Found

Content:

{
   "errors":"Loan **LOAN_ID** for **USER_ID** Not Found for broker **BROKER_EMAIL.**"
}

Sample Call

```
$.ajax({
  url: "/api/loan/cmpda8226579",
  dataType: "json",
  data : {
    "token":
    "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9
      e756c9525c0f44d3aaa7",
  },
  type : "GET",
  success : function(r) {
    console.log(r);
  }
});
```

SUBMIT LOAN REQUEST FOR SOFT PREQUAL

Title	Step 2B – Submit Loan Request for Soft Prequal
Description	The following API command submits a created loan request for soft credit prequalification. On success the command will return a prequal acceptance or rejection.
URL	/api/loans/LOAN_ID/prequals
Method	POST
URL Parameters	**Required:** loan_id=[string] example: cmpdb7897594
Data Parameters	N/A

Success Response	**Code:** 200 OK
	
	Content:
	
	`{`
	`  "loan":{`
	`        "amount": 35000,`
	`        "amount_approved": null,`
	`        "description": "some info",`
	`        "length_max": 6,`
	`        "status": "activated"`
	`  },`
	`  "prequal":{`
	`        "created_at": "2013/02/01 15:31:14 -0500",`
	`        "result": "accept",`
	`        "type": "DecissionPrescreen"`
	`  }`
	`}`

Error Response	**Code:** 409 Conflict
	
	Content:
	
	`{`
	`  "errors": {`
	`        "prequal":"Prequal Decision Already Made"`
	`  }`
	`}`

	Code: 412 Precondition Failed
	
	Content:
	
	`{`
	`  "errors": {`
	`        "prequal": [“Field 1 Error Message”, “Field 2 Error Message” , ...],`
	`  }`
	`}`

Sample Call	`$.ajax({`
	` url: "/api/loans/ cmpdb7897594/prequals",`
	` dataType: "json",`
	` data : {`
	`  "token":`
	`  "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9`
	`   e756c9525c0f44d3aaa7",`
	` },`
	` type : "POST",`
	` success : function(r) {`
	`  console.log(r);`
	` }`
	`});`

GET LOAN AGREEMENTS

Title	Get Loan Agreements
Description	The following API fetches a list of all available and signed Loan Agreements. On success the command will return a JSON representation of the available Loan Agreements. *Note: All agreements are available for pull after the loan offer has been signed and accepted by the borrower.*
URL	/api/loans/**LOAN_ID**/agreements/
Method	GET
URL Parameters	**Required:** loan_id=[string] example: cmpda8226579

Data Parameters

N/A

Success Response

Code: 200 OK

Content:

```
{
  "agreements":[
      {"kind":"terms_of_use", "created_at":"2013/01/11 14:32:43 -0500",
      "title":"Terms of Use Agreement", "id":1},
      {"kind":"promissory_note", "created_at":"2013/01/11 16:02:43 -0500",
      "title":"Promissory Note", "id":2},
      {"kind":"auto_fund", "created_at":"2013/01/11 16:02:43 -0500",
      "title":"Personaly Guaranty", "id":3},
      {"kind":"pad_borrower", "created_at":"2013/01/11 16:02:43 -0500",
      "title":"Authorization for Electronic Debits Agreement", "id":4},
  ]
}
```

Error Response

Code: 204 Not Content

Content:

```
{
  "errors":"No Agreements."
}
```

Sample Call

```
$.ajax({
 url: "/api/loan/cmpda8226579/agreements",
 dataType: "json",
 data : {
  "token":
  "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9
   e756c9525c0f44d3aaa7",
 },
 type : "GET",
 success : function(r) {
  console.log(r);
 }
});
```

GET LOAN AGREEMENT BODY

Title	Get Loan Agreement Body

Description	The following API fetches the body of a Loan Agreement. On success the command will return an embedded HTML representation of the Legal Agreement.
URL	/api/loans/**LOAN_ID**/agreements/**AGREEMENT_ID**
Method	GET
URL Parameters	**Required:** loan_id=[string] example: cmpda8226579 agreement_id=[number] example: 1
Data Parameters	N/A
Success Response	**Code:** 200 OK **Content:** HTML Agreement

Error Response	**Code:** 204 Not Content
	
	Content:
	
	`{`
	`  "errors":{"agreement":"No Agreement Found"}`
	`}`

Sample Call	`$.ajax({`
	` url: "/api/loan/cmpda8226579/agreements/1",`
	` dataType: "json",`
	` data : {`
	`  "token":`
	`  "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9`
	`   e756c9525c0f44d3aaa7",`
	` },`
	` type : "GET",`
	` success : function(r) {`
	`  console.log(r);`
	` }`
	`});`

UPLOAD ATTACHMENTS

Title	Upload Attachments
Description	The following API command creates/uploads a list of files to the loan application. Each file should have a list of document types appended to it. Every file uploaded should have a unique key name in the JSON data structure along with a matching item in the checklist parameter. **NOTE:** A loan application should have at minimum the 6-month bank statements attached to it.
URL	/api/loans/**LOAN_ID**/attachments
Method	POST
URL Parameters	**Required:** loan_id=[string] example: cmpda8226579

Data Parameters	• **files:** *Binary File, A list of binary files to upload.*
	
	• **checklist:** *A string representation of a data dictionary/hash, An ordered list of document types corresponding to each document uploaded. Accepted values:*
	○ *bank_statements*
	○ *void_check*
	○ *tax_returns*
	○ *drivers_license*
	○ *personal_guaranty*
	○ *credit_authorization*
	○ *marketing_agreement*
	○ *w9*
	○ *other*
	
	NOTE: The order of the document types in the file checklist parameter should match the exact file order in the corresponding previous parameter.
	
	<u>Example:</u>
	
	`{`
	`  "files":`
	`  {`
	`      "file1":"DRIVER LICENSE.JPEG",`
	`      "file2":"6 MONTH BANK STMT.PDF",`
	`      "file3":...`
	`  }`

```
   "checklist":"file1:[void_check, personal_guaranty], file2:[credit_authorization],
file3:[drivers_license]"
}
```

Success Response

Code: 200 OK

Content:

```
{
   "attachments":[
      {
         "size":188,
         "created_on":"2013/01/15 14:02:10 -0500",
         "content_type":"application/octet-stream",
         "id":14675,"filename":"DRIVER LICENSE.JPEG"
      },
      {
         "size":5922,
         "created_on":"2013/01/15 14:02:10 -0500",
         "content_type":"application/octet-stream",
         "id":14675,"filename":"6 MONTH BANK STMT.PDF"
      }
}
```

Error Response

Code: 400 Bad Request

Content:

```
{
   "errors": {
         "attachments": ["Attachment Error Messages", …],
   }
}
```

Sample Call

AJAX Example:

```
$.ajax({
 url: "/api/loans/cmpda8226579/attachments",
 dataType: "json",
 data : {
   "token":
   "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9
    e756c9525c0f44d3aaa7",
   "files":{
       "file1":/Users/IOU/files/file1.pdf,
       "file2":/Users/IOU/files/file2.jpeg
   }
   "checklist":
   "file1:[void_check, personal_guaranty], file2:[credit_authorization] "
 },
```

	```
	  type : "POST",
	  success : function(r) {
	    console.log(r);
	  }
	});
	```
	 CURL Example: `curl -X POST -F files[file1]=@'/Users/IOU/files/file1.pdf' -F files[file2]=@'/Users/IOU/files/file2.jpeg ' -F checklist='file1:[void_check, personal_guaranty], file2:[credit_authorization]' -F 'token=0b937faf333d6e3e8033a0c75907626145bfb7eeeca8e76be829f6f43650f543' http://@localhost:3000/api/loans/cmpDI49315487/attachments`

GET LOAN ATTACHMENTS

Title	Get Loan Attachments
Description	The following API fetches a list of all available Loan Attachments. On success the command will return a JSON representation of the available Loan Attachments.
URL	/api/loans/**LOAN_ID**/attachments/
Method	GET
URL Parameters	**Required:** loan_id=[string] example: cmpda8226579
Data Parameters	N/A

Success Response	**Code:** 200 OK **Content:**
	```
	{
	  "attachments": [
	    {
	      "content_type": "application/octet-stream",
	      "created_on": "2013/01/15 14:02:10 -0500",
	      "filename": "README_FOR_APP",
	      "id": 14675,
	      "size": 188
	    },
	```

	`{`
	`  "content_type": "image/gif",`
	`  "created_on": "2013/01/15 14:02:10 -0500",`
	`  "filename": "spinner.gif",`
	`  "id": 14676,`
	`  "size": 5922`
	`}`
	`]`
	`}`

Error Response	**Code:** 204 Not Content
	
	Content:
	
	`{`
	`  "errors":"No Attachments."`
	`}`

Sample Call	`$.ajax({`
	` url: "/api/loan/cmpda8226579/attachments",`
	` dataType: "json",`
	` data : {`
	`  "token":`
	`  "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9`
	`   e756c9525c0f44d3aaa7",`
	` },`
	` type : "GET",`
	` success : function(r) {`
	`  console.log(r);`
	` }`
	`});`

GET LOAN ATTACHMENT BODY

Title	Get Loan Attachment Body
Description	The following API fetches the body of a Loan Attachment. On success the command will return an embedded body of the Loan Attachment.
URL	/api/loans/**LOAN_ID**/attachments/**ATTACHMENT_ID**
Method	GET

URL Parameters	**Required:** loan_id=[string] example: cmpda8226579 attachment_id=[number] example: 14767
Data Parameters	N/A
Success Response	**Code:** 200 OK **Content:** Attachment Body/Content
Error Response	**Code:** 204 Not Content **Content:** `{ "errors":{"attachment":"No Attachment Found"} }`
Sample Call	`$.ajax({ url: "/api/loan/cmpda8226579/attachments/14676", dataType: "json", data : { "token": "86ed2fc3bdb25a0e297779bec49c1f55a2e84fd570a9 e756c9525c0f44d3aaa7", }, type : "GET", success : function(r) { console.log(r); } });`