
GoldStar Trust Company

CrowdPay Web API

Contents

Revisions	5
Introduction	8
URL	8
Example:.....	8
HTTP Headers.....	8
Security	9
Tools.....	9
Status Codes.....	10
Object Models.....	13
Investor	13
Account	15
Asset.....	17
Asset (Version 2)	18
Transaction (for Fund Account and Fund Escrow api function).....	19
Debt Transaction (for Fund Debt Escrow api function)	20
Debt Reinvest/Pay Transaction (for Debt Reinvest and Debt Pay api functions)	22
Debt Pay Transaction (for Debt Pay api functions).....	24
Escrow	25
EscrowRegister.....	26
EscrowTransaction.....	27
Investor	28
Create Investor.....	28
Request	28
Response	29
Get Investor	31
Request	31
Response.....	31
Account	33
Create Account.....	33
Request	33
Response.....	34
Get Account	35

Request	35
Response	35
Get Account with Assets	36
Request	36
Response	36
Get Account with Assets (Version 2).....	38
Request	38
Response	38
Get Account with Transactions	40
Request	40
Response	40
Update Account	42
Request	42
Response	43
Asset.....	44
Get Asset	44
Request	44
Response	44
Get Asset (Version 2)	44
Request	44
Response	44
Get Asset with Transactions	45
Request	45
Response	45
Get Asset with Transactions (Version 2)	46
Request	46
Response	46
Escrows	47
Get Escrows.....	47
Request	47
Response	47
Get Escrow	48
Request	48

Response	48
Get Register	49
Request	49
Response	49
Get Transactions	51
Request	51
Response	51
Transaction.....	52
Get Transaction.....	52
Request	52
Response	52
Fund Account	53
Request	54
Response	54
Fund Escrow	55
Request	55
Response	55
Fund Debt Escrow	56
Request	56
Response	57
Debt Reinvest.....	58
Request	58
Response	59
Debt Pay.....	60
Request	60
Response	61

Revisions

01/08/2015	<p>Added "investor_key" to Investor Model</p> <ul style="list-style-type: none"> a. Added "investor_key" to Response on "Create Investor" api. This "investor_key" must be used for subsequent "Get Investor Information" calls
01/11/2015	Added Api calls for Funding Account and Funding Escrow
01/22/2015	<p>Added ICollection<Asset> to Account Object Model</p> <p>Added addition information to FundEscrow api.</p> <p>Added quantity to FundEscrow api Request and Response sample.</p>
02/03/2015	<p>Added ICollection<Transaction> to Account model.</p> <p>Added Asset model.</p> <p>Added Transaction model.</p> <p>Added Get Account with Assets operation.</p> <p>Added Get Account with Transactions operation.</p> <p>Added Get Asset operation.</p> <p>Added Get Asset with Transactions operation.</p>
02/06/2015	<p>Added Postman examples to Create Investor and Get Investor operations.</p> <p>Formatting changes.</p>
02/11/2015	<p>Modified Account model</p> <ul style="list-style-type: none"> a. Added ideology_id b. Added contact_name c. Added contact_phone d. Added contact_email e. Added "contact_name": "InvestorFirst InvestorLast", to Create Account sample content to POST f. Added "contact_phone": "8061234567", to Create Account sample content to POST g. Added "contact_email": "fi+rst.l+ast@somewhere.com", to Create Account sample content to POST h. Added "idology id:" 1234567 to Create Account sample content to POST
02/26/2015	<p>Modified Investor model</p> <ul style="list-style-type: none"> a. Added created_by_ip_address as required property <p>Modified Account model</p> <ul style="list-style-type: none"> a. Added created_by_ip_address as required property <p>Modified Transaction model</p> <ul style="list-style-type: none"> a. Added created_by_ip_address as required property <p>Modified Create Investor endpoint</p> <ul style="list-style-type: none"> a. Added required propert created_by_ip_address <p>Modified Create Account endpoint</p> <ul style="list-style-type: none"> a. Added required propert created_by_ip_address <p>Modified Fund Account endpoint</p> <ul style="list-style-type: none"> a. Added required propert created_by_ip_address

	<p>Modified Fund Escrow endpoint</p> <ul style="list-style-type: none"> a. Added required property created_by_ip_address <p>Added Escrow endpoint</p> <ul style="list-style-type: none"> a. Get accounts b. Get transactions
03/12/2015	<p>Added Models</p> <ul style="list-style-type: none"> a. Escrow b. EscrowRegister c. EscrowTransaction <p>Modified Escrow endpoints</p> <ul style="list-style-type: none"> d. Get escrows e. Get escrow by id f. Get escrow by id with register g. Get escrow by id with transactions
03/20/2015	Added UpdateAccount api
04/15/2015	<p>Added Model</p> <ul style="list-style-type: none"> • Debt Transaction (for Fund Debt Escrow api) <p>Added Fund Debt Escrow endpoint - https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction/FundDebtEscrow</p>
05/07/2015	<p>Added Version 2 Asset Model</p> <p>Added Version 2 endpoints for Getting Assets created by Fund Debt Escrow api</p> <ul style="list-style-type: none"> a. https://qaservices.goldstartrust.com/Crowdfunding/api/V2/Account/account_id/Assets b. https://qaservices.goldstartrust.com/Crowdfunding/api/V2/Account/account_id/Assets/asset_id c. https://qaservices.goldstartrust.com/Crowdfunding/api/V2/Account/account_id/Assets/asset_id/transactions
05/15/2015	Added url endpoints to return “All” assets and/or transactions for an account at the api level. Prior endpoints would return based on api key and portal key.
05/19/2015	<p>Added Status Code section – Lists Http response codes with possible messages</p> <p>Added previously undocumented api call - Get Transaction</p>
05/26/2015	<p>Breaking Changes</p> <p>Investor Model - the following fields are now required</p> <ul style="list-style-type: none"> • If is_person = true, then first_name and last_name are required • If is_person = false, the name is required • legal_address_1, city, state, zip <p>Account Model – the following fields are now required</p> <p>contact_email</p>
06/03/2015	<p>Added url endpoints Debt reinvestment, posting interest and principal to debt assets as well as selling/maturing a debt asset</p> <ul style="list-style-type: none"> • Added models

	<ul style="list-style-type: none"> ○ Debt Reinvest transaction for new url endpoint below ○ Debt Pay transaction for new url endpoint below • https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction/ReinvestDebt • https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction/DebtPay

Introduction

The CrowdPay API is a REST-based interface to our CrowdPay Escrow services. Before making API calls, you will need to contact GoldStar Trust to receive credentials and get your server IP addresses whitelisted.

URL

The URL schema for the API is:

```
Version 1
https://{domain}/crowdfunding/api/{resource}

Version 2 and beyond
https://{domain}/crowdfunding/api/{version}/{resource}
```

The meaning of each token is:

Token	Description
{domain}	You will receive the domain name at the time you receive your credentials.
{version}	The version you want to communicate with (ex. v1 or v2).
{resource}	The resource you want to perform operations on.

Example:

The endpoint **POST** <https://qaservices.goldstartrust.com/crowdfunding/api/Investor> with the Investor model posted will create an Investor resource.

HTTP Headers

Header	Description
X-ApiKey	You will receive your api key from Goldstar Trust. All api calls will require this header.
X-PortalKey	You will receive your portal key from Goldstar Trust. All api calls will require this header. – This identifies the escrow (offering).
X-ByPassValidation	Set to True or False. Only functional in QA (sandbox). It will be ignored in Production. This header is used to bypass production validation that would normally prevent transactional processing. <ol style="list-style-type: none">1. Allows you to execute FundEscrow transactions regardless of account available balance. In production and when value is set to False or header is not passed in, FundEscrow transaction will be rejected if amount is greater than available balance. Normal workflow is:<ul style="list-style-type: none">○ FundAccount transaction is initiated.

	<ul style="list-style-type: none"> ○ After 3 work days, the account's available balance reflects the amount of the FundAccount transaction. ○ The FundEscrow calls will now accept transactions less than or equal to account's available balance. <p>2. Returns status of "Processed" for transactions when querying transactions. Normally, the status would be "Pending" until the transactions were posted by GoldStar.</p>

Security

We require all calls to be made securely with HTTP over SSL.

All Api calls will require the HTTP header X-ApiKey and X-PortalKey listed above.

Tools

We like using **Postman** when working with the API.

Status Codes

Response will have one of the following status codes.

Code	Description
200	Success – requested resource is in the response.
201	Success – the newly created resource is in the response.
400	<p>Bad Request – could not be understood by the server. API call failed.</p> <p>Possible reasons:</p> <ul style="list-style-type: none"> ○ Requested method or operation is not implemented (e.g. trying to do a POST for an endpoint that does not support a POST) ○ Unhandled exception – An exception thrown by the api that the code is not handling. Contact GoldStar so that we can investigate
401	<p>Unauthorized – requested resource requires authentication.</p> <p>Error Code 80000 - Unauthorized Access to Account</p> <p>Error code 80002 – Unauthorized Access to Account – Account is NOT setup for Crowdfunding (Can occur for any of the api calls below)</p> <p>Error Code 89999 – Unauthorized access to api. Missing credentials. Can occur for any of the api calls below.</p> <ul style="list-style-type: none"> ● Create Account ● Get Account <ul style="list-style-type: none"> ○ Error code 80003 – Unauthorized Access to Account – Unauthorized Access to Account - Account is not associated with Portal/Escrow ● Get Account with Assets <ul style="list-style-type: none"> ○ Error code 80003 – Unauthorized Access to Account – Unauthorized Access to Account - Account is not associated with Portal/Escrow ● Get Account with Asset using asset_id <ul style="list-style-type: none"> ○ Error code 80003 – Unauthorized Access to Account – Unauthorized Access to Account - Account is not associated with Portal/Escrow ● Get Account with Transactions <ul style="list-style-type: none"> ○ Error code 80003 – Unauthorized Access to Account – Unauthorized Access to Account - Account is not associated with Portal/Escrow ● Get Transaction using transaction_id ● Fund Account <ul style="list-style-type: none"> ○ 80001 - Unauthorized Access to Account - Account set up has not passed CIP ○ Error code 80003 – Unauthorized Access to Account – Unauthorized Access to Account - Account is not associated with Portal/Escrow ● Fund Escrow <ul style="list-style-type: none"> ○ Error code 80003 – Unauthorized Access to Account – Unauthorized Access to Account - Account is not associated with Portal/Escrow ● Fund Deb Escrow <ul style="list-style-type: none"> ○ Error code 80003 – Unauthorized Access to Account – Unauthorized Access to Account - Account is not associated with Portal/Escrow

Code	Description
	<ul style="list-style-type: none"> • Reinvest Debt <ul style="list-style-type: none"> ○ Error code 80003 – Unauthorized Access to Account – Unauthorized Access to Account - Account is not associated with Portal/Escrow • Update Account <ul style="list-style-type: none"> ○ Error code 80003 – Unauthorized Access to Account – Unauthorized Access to Account - Account is not associated with Portal/Escrow
404	<p>Not Found – 10000 - requested resource does not exist on the server.</p> <ul style="list-style-type: none"> • Get Investor • Get Account • Get Account with Assets • Get Account with Asset using asset_id • Get Account with Transactions • Get Transaction using transaction_id • Fund Account • Fund Escrow • Fund Deb Escrow • Create Account – additional possible reasons <ul style="list-style-type: none"> ○ Investor record not found using investor_id provided in the model ○ Bank routing number invalid. Unable to find Bank record • Update Account <ul style="list-style-type: none"> ○ Investor record not found using investor_id provided in the model • Reinvest Debt <ul style="list-style-type: none"> ○ Asset Not found with provided id
409	<p>Validation Error</p> <ul style="list-style-type: none"> • Create Investor – 200000 - Investor with same tax id already exists. • Fund Escrow – numeric error code with error message returned in response <ul style="list-style-type: none"> ○ 85000 - Unauthorized Transaction - Account does not have sufficient funds for transaction (Available Balance) • Fund Debt Escrow – numeric error code with error message returned in response <ul style="list-style-type: none"> ○ 86000 - Fund Escrow Not set up correctly - Please contact Goldstar Trust ○ 86001 - Invalid Operation - Escrow is not set up as Debt Offering - Please contact Goldstar Trust • Reinvest Debt – numeric error code with error message returned in response <ul style="list-style-type: none"> ○ 86000 - Fund Escrow Not set up correctly - Please contact Goldstar Trust ○ 86001 - Invalid Operation - Escrow is not set up as Debt Offering - Please contact Goldstar Trust ○ 86002 - Invalid Transaction - Amount must equal zero or cost basis amount – This is when activity_code = 707

Code	Description
	<ul style="list-style-type: none"> ○ 86003 - Invalid Transaction - Amount must be less than cost basis amount – This occurs when activity_code = 708 ○ 86004 - Invalid Transaction - Asset is not active
500	Internal Server Error – Generic error occurred on the server. – API called failed. Can occur with any api call.

Object Models

Investor

```
public long id { get; set; }

public Guid? Investor_key { get; set; }

[StringLength(9)]
[MinLength(9)]
[MaxLength(9)]
[Required]
public string tax_id_number { get; set; }

[StringLength(50)]
public string first_name { get; set; }

[StringLength(50)]
public string middle_name { get; set; }

[StringLength(50)]
public string last_name { get; set; }

[StringLength(150)]
public string name { get; set; }

[Column(TypeName = "date")]
public DateTime? birth_date { get; set; }

[StringLength(40)]
public string mailing_address_1 { get; set; }

[StringLength(40)]
public string mailing_address_2 { get; set; }

[StringLength(40)]
public string mailing_city { get; set; }

[StringLength(30)]
public string mailing_state { get; set; }
[StringLength(9)]
public string mailing_zip { get; set; }

[StringLength(40)]
public string mailing_country { get; set; }

[Required]
public bool is_mailing_address_foreign { get; set; }

[StringLength(40)]
public string legal_address_1 { get; set; }

[StringLength(40)]
public string legal_address_2 { get; set; }

[StringLength(40)]
public string legal_city { get; set; }

[StringLength(30)]
```

```

public string legal_state { get; set; }

[StringLength(9)]
public string legal_zip { get; set; }

[StringLength(40)]
public string legal_country { get; set; }

public bool is_legal_address_foreign { get; set; }

[StringLength(10)]
public string primary_phone { get; set; }

[StringLength(10)]
public string secondary_phone { get; set; }

[Required]
public bool is_person { get; set; }

[StringLength(50)]
public string email { get; set; }

[Required]
public bool is_cip_satisfied { get; set; }

[StringLength(30)]
public string portal_investor_number { get; set; }

[StringLength(45)]
[Required]
public string created_by_ip_address { get; set; }

public ICollection<Account> Accounts

```

Account

If name_1, name_2, name_3 and name_4 are left null, Investor's name information will be used to value these properties.

```
public long id { get; set; }

[StringLength(15)]
public string number { get; set; }

[StringLength(30)]
public string portal_account_number { get; set; }

[Required]
public long investor_id { get; set; }

[StringLength(50)]
public string name_1 { get; set; }

[StringLength(50)]
public string name_2 { get; set; }

[StringLength(50)]
public string name_3 { get; set; }

[StringLength(50)]
public string name_4 { get; set; }

[StringLength(40)]
public string mailing_address_1 { get; set; }

[StringLength(40)]
public string mailing_address_2 { get; set; }

[StringLength(40)]
public string mailing_city { get; set; }

[StringLength(30)]
public string mailing_state { get; set; }

[StringLength(9)]
public string mailing_zip { get; set; }

[StringLength(40)]
public string mailing_country { get; set; }

[Required]
public bool is_mailing_address_foreign { get; set; }

public decimal? current_uninvested_cash { get; set; }

public decimal? uninvested_cash { get; set; }

[Required]
public bool is_cip_satisfied { get; set; }
```

```

// 1=Checking, 2=Savings
[Required]
[RegularExpression(@"[1-2]")]
public int? draft_account_type_id { get; set; }

[Required]
[MinLength(9)]
[MaxLength(9)]
public string draft_routing_number { get; set; }

[Required]
[StringLength(17)]
public string draft_account_number { get; set; }

[Required]
[StringLength(50)]
public string draft_account_name { get; set; }

// 1=Open, 2=Closed, 3=Pending
[Required]
[RegularExpression(@"[1-3]")]
public int? status_id { get; set; }

// 2=Traditional IRA, 11=Roth IRA 12=Non IRA
[Required]
[RegularExpression(@"[2]|1[1-2]")] // Valid values 2, 11, 12
public int account_type_id { get; set; }

// 1=Received W-9/No Withholding           2=W-9 Not Received/Withholding
// 3=Mandatory Withholding                 4=Exempt From Reporting
// 5=Corporate - No Reporting               6=Foreign Resident
// 7=W-8 Non US Citizen
[RegularExpression(@"[1-7]")] // Valid values 1 thru 7
[Required]
public int? w9_code_id { get; set; }

[StringLength(75)]
public string contact_name { get; set; }

[StringLength(20)]
[RegularExpression(@"^\d+$")]
public string contact_phone { get; set; }

[StringLength(50)]
[Required]
public string contact_email { get; set; }

public long? ideology_id { get; set; }

[StringLength(45)]
[Required]
public string created_by_ip_address { get; set; }

public ICollection<Asset> Assets

public ICollection<Transaction> Transactions

```



```
Asset
public long id
public string description
public string number
public DateTime? sold_date
public decimal? market_value
public ICollection<Transaction> Transactions
```

Asset (Version 2)

```
public long id

[StringLength(100)]
public string description

[StringLength(35)]
public string number

public DateTime? sold_date

public decimal? market_value

[RegularExpression(@"^([0-9]{0,2})(\.[0-9]{1})?$", ErrorMessage = "Invalid term")]
public decimal? term { get; set; }

public DateTime? effective_date

[RegularExpression(@"^[1-2]", ErrorMessage = "Invalid interest type. 1=Simple, 2=Compound")]
public decimal? interest_type { get; set; }

[RegularExpression(@"^[1-4]", ErrorMessage = "Invalid interest frequency. 1=Monthly, 2=Quarterly, 3=Semi-Annually, 4=Annually")]
public decimal? interest_frequency { get; set; }

[Required]
[Range(0.000, 99.999, ErrorMessage = "Interest rate must be between 0.000 and 99.999 inclusively.")]
[RegularExpression(@"^([0-9]{0,2})(\.[0-9]{1,3})?$", ErrorMessage = "Invalid interest_rate")]
public virtual decimal interest_rate

public DateTime? maturity_date

[StringLength(30)]
public string third_party_asset_number

public ICollection<Transaction> Transactions
```

Transaction (for Fund Account and Fund Escrow api function)

```
public long id

[Required]
public long account_id

public long? asset_id

public DateTime? date

[StringLength(20)]
public String reference

[StringLength(50)]
public String description

[Required]
public virtual decimal amount

// Pending, Processed
public string status

[StringLength(45)]
[Required]
public string created_by_ip_address { get; set; }
```

Debt Transaction (for Fund Debt Escrow api function)

```
public long id

[Required]
public long account_id

public long? asset_id

public DateTime? date

[StringLength(20)]
public String reference

[StringLength(50)]
public String description

[Required]
[Range(0.01, 999999999999.99, ErrorMessage = "Amount must be greater than 0.00")]
[RegularExpression(@"\d+(\.\d{1,2})?", ErrorMessage = "Invalid amount")]
public decimal amount

// Pending, Processed
public string status

[StringLength(45)]
[Required]
public string created_by_ip_address { get; set; }

[RegularExpression(@"^([0-9]{0,2})(\.[0-9]{1})?$", ErrorMessage = "Invalid term")]
public decimal? term { get; set; }

public DateTime? effective_date

[RegularExpression(@"[1-2]", ErrorMessage = "Invalid interest type. 1=Simple, 2=Compound")]
public decimal? interest_type { get; set; }

[RegularExpression(@"[1-4]", ErrorMessage = "Invalid interest frequency. 1=Monthly, 2=Quarterly, 3=Semi-Annually, 4=Annually")]
public decimal? interest_frequency { get; set; }

[Required]
[Range(0.000, 99.999, ErrorMessage = "Interest rate must be between 0.000 and 99.999 inclusively.")]
[RegularExpression(@"^([0-9]{0,2})(\.[0-9]{1,3})?$", ErrorMessage = "Invalid interest_rate")]
public decimal interest_rate

[Required]
public DateTime? maturity_date

[StringLength(30)]
public string third_party_asset_number

[StringLength(100)]
public string asset_description
```

```
[StringLength(30)]  
public string cusip_number
```

Debt Reinvest/Pay Transaction (for Debt Reinvest and Debt Pay api functions)

```
public long id

[Required]
public long account_id

public long? asset_id

public DateTime? date

[StringLength(20)]
public String reference

[StringLength(50)]
public String description

[Required]
[Range(0.00, 999999999999.99, ErrorMessage = "Amount must be greater than or equal 0.00")]
[RegularExpression(@"\d+(\.\d{1,2})?", ErrorMessage = "Invalid amount")]
public decimal amount

// Pending, Processed
public string status

[StringLength(45)]
[Required]
public string created_by_ip_address { get; set; }

// 707 - Total Reinvestment, 708 - Partial Reinvestment
[Required]
[RegularExpression(@"^707|708$")]
public string activity_code { get; set; }

public DateTime? sold_date { get; set; }

[RegularExpression(@"^([0-9]{0,2})(\.[0-9]{1})?$", ErrorMessage = "Invalid term")]
public decimal? term { get; set; }

public DateTime? effective_date

[RegularExpression(@"[1-2]", ErrorMessage = "Invalid interest type. 1=Simple, 2=Compound")]
public decimal? interest_type { get; set; }

[RegularExpression(@"[1-4]", ErrorMessage = "Invalid interest frequency. 1=Monthly, 2=Quarterly, 3=Semi-Annually, 4=Annually")]
public decimal? interest_frequency { get; set; }

[Required]
[Range(0.000, 99.999, ErrorMessage = "Interest rate must be between 0.000 and 99.999 inclusively.")]
[RegularExpression(@"^([0-9]{0,2})(\.[0-9]{1,3})?$", ErrorMessage = "Invalid interest_rate")]
public decimal interest_rate

[Required]
```

```
public DateTime? maturity_date

[StringLength(30)]
public string third_party_asset_number

[StringLength(100)]
public string asset_description

[StringLength(30)]
public string cusip_number
```

Debt Pay Transaction (for Debt Pay api functions)

```
public long id

[Required]
public long account_id

public long? asset_id

public DateTime? date

[StringLength(20)]
public String reference

[StringLength(50)]
public String description

[Required]
[Range(0.00, 999999999999.99, ErrorMessage = "Amount must be greater than or equal
0.00")]
[RegularExpression(@"\d+(\.\d{1,2})?", ErrorMessage = "Invalid amount")]
public decimal amount

// Pending, Processed
public string status

[StringLength(45)]
[Required]
public string created_by_ip_address { get; set; }

// 301 - Interest Income, 304 - Non-cash dividend (increases principal),
// 305 - Asset Sold/Matured
[Required]
[RegularExpression(@"^301|304|305$")]
public string activity_code { get; set; }

public DateTime? sold_date { get; set; }
```


Escrow

```
public long id { get; set; }

public string issue_number { get; set; }

public string portal_issue_number { get; set; }

public string offering_type { get; set; }

public decimal minimum_investment_amount { get; set; }

public decimal maximum_investment_amount { get; set; }

public decimal issue_amount { get; set; }

public decimal cash_balance { get; set; }

public decimal principal_balance { get; set; }

public ICollection<EscrowRegister> Register { get; set; }

public ICollection<EscrowTransaction> Transactions { get; set; }
```

EscrowRegister

```
public long account_id { get; set; }  
public long asset_id { get; set; }  
public string asset_number { get; set; }  
public DateTime date { get; set; }  
public decimal quantity { get; set; }  
public decimal amount { get; set; }  
public string status { get; set; }
```

EscrowTransaction

```
public DateTime date { get; set; }  
public string description { get; set; }  
public decimal amount { get; set; }  
public string status { get; set; }
```

Investor

Create Investor

Creates a new investor resource. The `id` and `investor_key` attributes are assigned by the system and returned in Response payload. Subsequent calls to GET the investor must use the `investor_key`.

For compliance reasons, the following properties will be required.

- if `is_person` = true, `first_name` and `last_name` is required
- if `is_person` = false, `name` is required
- `legal_address_1`, `legal_city`, `legal_state`, `legal_zip`

Request

Content-Type: application/json

POST <https://qaservices.goldstartrust.com/Crowdfunding/api/Investor>

```
{
  "id": 0,
  "tax_id_number": "111223334",
  "first_name": "InvestorFirst",
  "middle_name": "InvestorMiddle",
  "last_name": "InvestorLast",
  "name": null,
  "birth_date": "1960-12-31T00:00:00",
  "mailing_address_1": "123 Ave A",
  "mailing_address_2": null,
  "mailing_city": "Somewhere",
  "mailing_state": "TX",
  "mailing_zip": "79109",
  "mailing_country": null,
  "is_mailing_address_foreign": false,
  "legal_address_1": "123 Ave A",
  "legal_address_2": null,
  "legal_city": "Soemwhere",
  "legal_state": "TX",
  "legal_zip": "79109",
  "legal_country": null,
  "is_legal_address_foreign": false,
  "primary_phone": "1112223333",
  "secondary_phone": "2223334444",
  "is_person": true,
  "email": "first.last@somewhere.com",
  "is_cip_satisfied": false,
  "portal_investor_number": "yourinvestornumber",
  "created_by_ip_address": "123.456.789.012"
}
```

Response

Notice the payload includes the system-assigned `id` and `investor_key` attributes. The `id` attribute must be used when creating accounts associated with this investor. The `investor_key` attribute must be used to GET this investor.

```
{
  "id": 76866,
  "investor_key": "abababab-cdcd-efef-ghgh-jkjkjkjkjkjk",
  "tax_id_number": "111223334",
  "first_name": "InvestorFirst",
  "middle_name": "InvestorMiddle",
  "last_name": "InvestorLast",
  "name": null,
  "birth_date": "1960-12-31T00:00:00",
  "mailing_address_1": "123 Ave A",
  "mailing_address_2": null,
  "mailing_city": "Somewhere",
  "mailing_state": "TX",
  "mailing_zip": "79109",
  "mailing_country": null,
  "is_mailing_address_foreign": false,
  "legal_address_1": "123 Ave A",
  "legal_address_2": null,
  "legal_city": "Soemwhere",
  "legal_state": "TX",
  "legal_zip": "79109",
  "legal_country": null,
  "is_legal_address_foreign": false,
  "primary_phone": "1112223333",
  "secondary_phone": "2223334444",
  "is_person": true,
  "email": "first.last@somewhere.com",
  "is_cip_satisfied": false,
  "portal_investor_number": "yourinvestornumber",
  "created_by_ip_address": "123.456.789.012",
  "Accounts": [],
  "Transactions": []
}
```

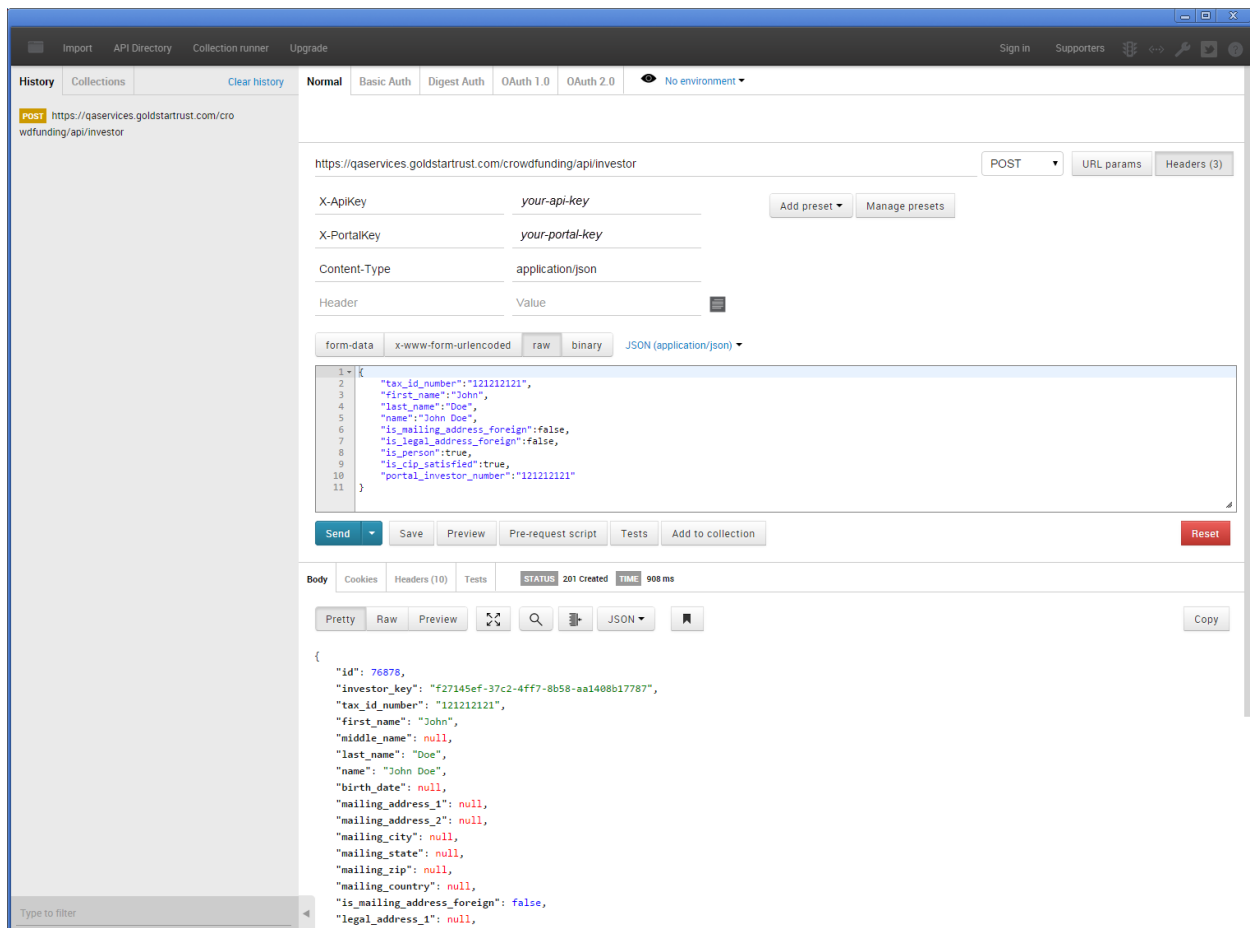


Figure 1 - Create investor example using Postman.

Get Investor

Get information on investor specified by the `investor_key` attribute. Will include all accounts associated to the investor.

Request

GET https://qaservices.goldstarttrust.com/Crowdfunding/api/Investor/investor_key

Response

```
{
  "id": 123456,
  "investor_key": "abababab-cdcd-efef-ghgh-jkjkjkjkjkjk",
  "tax_id_number": "111223333",
  "first_name": "InvestorFirst",
  "middle_name": "InvestorMiddle",
  "last_name": "InvestorLast",
  "name": null,
  "birth_date": "1960-12-31T00:00:00",
  "mailing_address_1": "123 Ave A",
  "mailing_address_2": null,
  "mailing_city": "Somewhere",
  "mailing_state": "TX",
  "mailing_zip": "79109",
  "mailing_country": null,
  "is_mailing_address_foreign": false,
  "legal_address_1": "123 Ave A",
  "legal_address_2": null,
  "legal_city": "Soemwhere",
  "legal_state": "TX",
  "legal_zip": "79109",
  "legal_country": null,
  "is_legal_address_foreign": false,
  "primary_phone": "1112223333",
  "secondary_phone": "2223334444",
  "is_person": true,
  "email": "first.last@somewhere.com",
  "is_cip_satisfied": false,
  "portal_investor_number": "yourinvestornumber",
  "created_by_ip_address": "123.456.789.012",    "Accounts": []
}
```

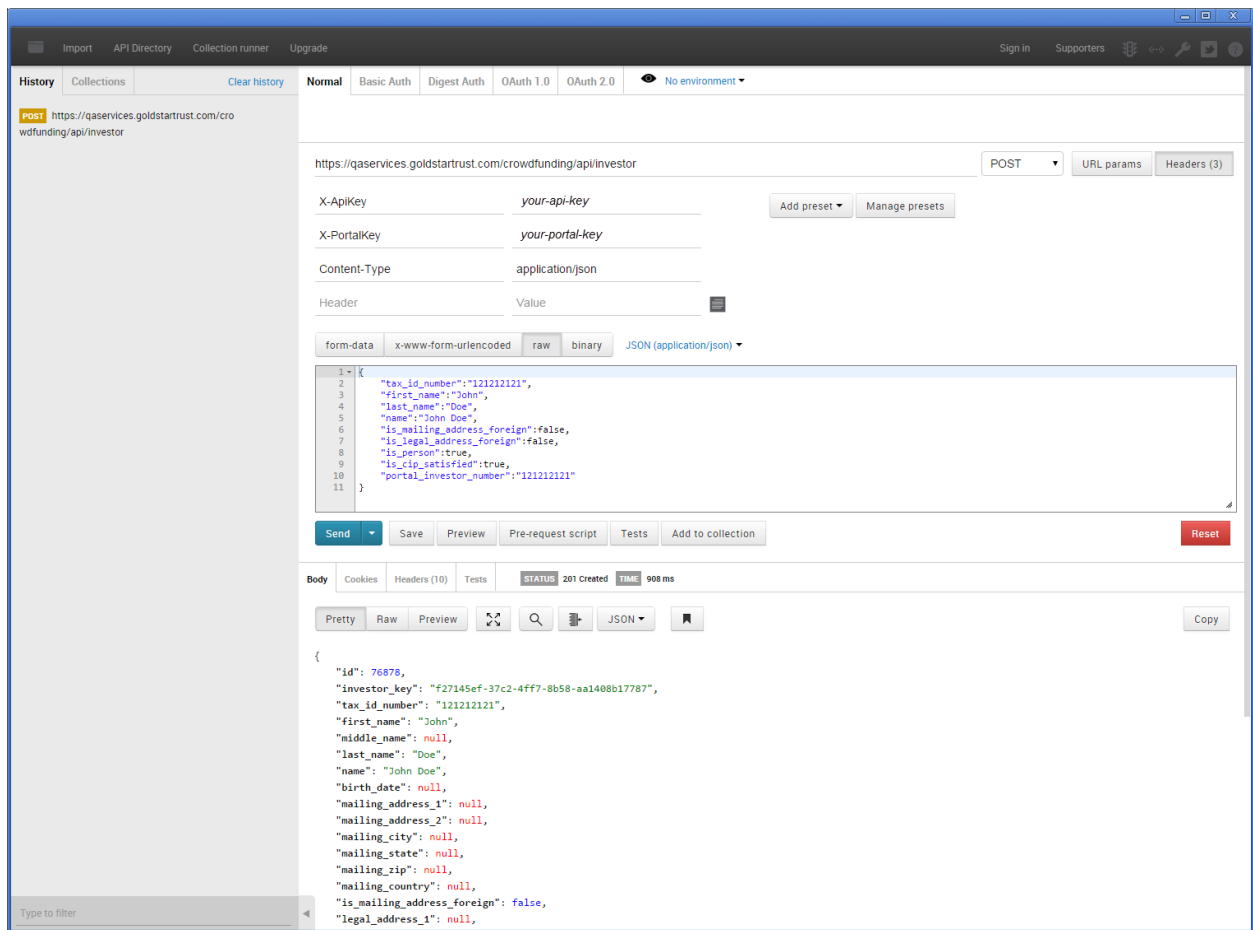


Figure 2 – Get investor example using Postman.

Account

Create Account

Creates Investor Account record – id returned must be used for subsequent “Funding” and “Investment” API calls. Make sure `investor_id` attribute is populated. This is what is used to tie account to investor.

For compliance reasons, the following properties will be required.

- `contact_email`

If `name_1`, `name_2`, `name_3` and `name_4` attributes are null, the investor’s name information will be used to value the name properties.

If `mailing_address_1`, `mailing_address_2`, `mailing_city`, `mailing_state`, `mailing_zip`, and `mailing_country` attributes are null, the investor’s address information will be used to value the account address properties.

Request

Content-Type: application/json

POST <https://qaservices.goldstarttrust.com/Crowdfunding/api/Account>

```
{
  "portal_account_number": "youraccountnumber",
  "investor_id": 76866,
  "name_1": null,
  "name_2": null,
  "name_3": null,
  "name_4": null,
  "mailing_address_1": "123 Ave A",
  "mailing_address_2": null,
  "mailing_city": "Somewhere",
  "mailing_state": "TX",
  "mailing_zip": "79109",
  "mailing_country": null,
  "is_mailing_address_foreign": "false",
  "is_cip_satisfied": true,
  "draft_account_type_id": 1,
  "draft_routing_number": "111310870",
  "draft_account_number": "1234567890",
  "draft_account_name": "First Last",
  "status_id": 1,
  "account_type_id": 12,
  "w9_code_id": 1,
  "contact_name": "InvestorFirst InvestorLast",
  "contact_phone": "8061234567",
  "contact_email": "fi+rst.l+ast@somewhere.com",
  "idology_id": 123456,
  "created_by_ip_address": "123.456.789.012"
}
```

Response

Account Model is returned. Notice the `id` attribute returned; it must be used when making “Funding” and “Escrow” API calls.

```
{
  "id": 81686,
  "number": "$1000604176",
  "portal_account_number": "9876431",
  "investor_id": 76866,
  "name_1": "InvestorFirst InvestorMiddle InvestorLast",
  "name_2": null,
  "name_3": null,
  "name_4": null,
  "mailing_address_1": "123 Ave A",
  "mailing_address_2": null,
  "mailing_city": "Somewhere",
  "mailing_state": "TX",
  "mailing_zip": "79109",
  "mailing_country": null,
  "is_mailing_address_foreign": false,
  "current_uninvested_cash": 0,
  "uninvested_cash": 0,
  "is_cip_satisfied": false,
  "draft_account_type_id": null,
  "draft_routing_number": null,
  "draft_account_number": "1234567890",
  "draft_account_name": "First Last",
  "status_id": 1,
  "account_type_id": 12,
  "w9_code_id": 1,
  "contact_name": "InvestorFirst InvestorLast",
  "contact_phone": "8061234567",
  "contact_email": "fi+rst.l+ast@somewhere.com",
  "idology_id": 123456,
  "created_by_ip_address": "123.456.789.012"
}
```

Get Account

Get information on account specified by the `account_id` parameter.

Request

GET https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id

Response

```
{
  "id": 81690,
  "number": "$1000604180",
  "portal_account_number": "9876431",
  "investor_id": 123456,
  "name_1": "InvestorFirst InvestorMiddle InvestorLast",
  "name_2": null,
  "name_3": null,
  "name_4": null,
  "mailing_address_1": "123 Ave. A",
  "mailing_address_2": null,
  "mailing_city": "Amarillo",
  "mailing_state": "TX",
  "mailing_zip": "79109",
  "mailing_country": null,
  "is_mailing_address_foreign": false,
  "available_balance": 0,
  "current_balance": 0,
  "is_cip_satisfied": false,
  "draft_account_type_id": 1,
  "draft_routing_number": "111222333",
  "draft_account_number": "1234567890",
  "draft_account_name": "InvestorFirst InvestorLast",
  "status_id": 1,
  "account_type_id": 12,
  "w9_code_id": 1,
  "contact_name": "InvestorFirst InvestorLast",
  "contact_phone": "806-123-4567",
  "contact_email": "fi+rst.l+ast@somewhere.com",
  "idology_id": 123456,
  "created_by_ip_address": "123.456.789.012",
  "Assets": [],
  "Transactions": []
}
```

Get Account with Assets

Get information on account specified by the **account_id** parameter along with all assets for the account. The “All” keyword will return all assets for the account at the api key level. The original GET continues to return assets for the account at the **api key** and **portal key** level.

Request

GET https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Assets

GET https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Assets/All

Response

```
{
  "id": 81690,
  "number": "$1000604180",
  "portal_account_number": "9876431",
  "investor_id": 123456,
  "name_1": "InvestorFirst InvestorMiddle InvestorLast",
  "name_2": null,
  "name_3": null,
  "name_4": null,
  "mailing_address_1": "3700 Rutson Drive test",
  "mailing_address_2": null,
  "mailing_city": "Amarillo",
  "mailing_state": "TX",
  "mailing_zip": "79109",
  "mailing_country": null,
  "is_mailing_address_foreign": false,
  "available_balance": 0,
  "current_balance": 0,
  "is_cip_satisfied": false,
  "draft_account_type_id": 1,
  "draft_routing_number": "111222333",
  "draft_account_number": "1234567890",
  "draft_account_name": "InvestorFirst InvestorLast",
  "status_id": 1,
  "account_type_id": 12,
  "w9_code_id": 1,
  "contact_name": "InvestorFirst InvestorLast",
  "contact_phone": "806-123-4567",
  "contact_email": "fi+rst.l+ast@somewhere.com",
  "idology_id": 123456,
  "created_by_ip_address": "123.456.789.012",
  "Assets": [
    {
      "id": 471277,
      "description": "ACME Wealth Agriculture 101, LLC",
      "number": "2000001",
      "sold_date": null,
      "market_value": 250,
      "Transactions": []
    }
  ]
}
```

```
    }  
  ],  
  "Transactions": []  
}
```

Get Account with Assets (Version 2)

Same as Original version except that Asset model will be the version 2 model for Asset.

Request

GET https://qaservices.goldstartrust.com/Crowdfunding/api/V2/Account/account_id/Assets

GET https://qaservices.goldstartrust.com/Crowdfunding/api/V2/Account/account_id/Assets/All

Response

```
{
  "id": 81690,
  "number": "$1000604180",
  "portal_account_number": "9876431",
  "investor_id": 123456,
  "name_1": "InvestorFirst InvestorMiddle InvestorLast",
  "name_2": null,
  "name_3": null,
  "name_4": null,
  "mailing_address_1": "3700 Rutson Drive test",
  "mailing_address_2": null,
  "mailing_city": "Amarillo",
  "mailing_state": "TX",
  "mailing_zip": "79109",
  "mailing_country": null,
  "is_mailing_address_foreign": false,
  "available_balance": 0,
  "current_balance": 0,
  "is_cip_satisfied": false,
  "draft_account_type_id": 1,
  "draft_routing_number": "111222333",
  "draft_account_number": "1234567890",
  "draft_account_name": "InvestorFirst InvestorLast",
  "status_id": 1,
  "account_type_id": 12,
  "w9_code_id": 1,
  "contact_name": "InvestorFirst InvestorLast",
  "contact_phone": "806-123-4567",
  "contact_email": "fi+rst.l+ast@somewhere.com",
  "idology_id": 123456,
  "created_by_ip_address": "123.456.789.012",
  "Assets": [
    {
      "id": 471277,
      "description": "ACME Wealth Agriculture 101, LLC",
      "number": "2000001",
      "sold_date": null,
      "market_value": 250,
      "term": null,
      "effective_date": "2015-04-15T00:00:00",
      "interest_type": "Simple",
      "interest_frequency": "Monthly",

```

```
    "interest_rate": 3.25,  
    "maturity_date": "2017-10-15T00:00:00",  
    "third_party_asset_number": "yourassetnumber",  
    "quantity": 1,  
    "Transactions": []  
  },  
  ],  
  "Transactions": []  
}
```

Get Account with Transactions

Get information on account specified by the **account_id** parameter along with all transactions for the account. The "All" keyword will return all transactions for the account at the api key level. The original GET continues to return transactions for the account at the **api key** and **portal key** level.

Request

GET https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction

GET https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction/All

Response

```
{
  "id": 81690,
  "number": "$1000604180",
  "portal_account_number": "9876431",
  "investor_id": 123456,
  "name_1": "InvestorFirst InvestorMiddle InvestorLast",
  "name_2": null,
  "name_3": null,
  "name_4": null,
  "mailing_address_1": "3700 Rutson Drive test",
  "mailing_address_2": null,
  "mailing_city": "Amarillo",
  "mailing_state": "TX",
  "mailing_zip": "79109",
  "mailing_country": null,
  "is_mailing_address_foreign": false,
  "available_balance": 0,
  "current_balance": 0,
  "is_cip_satisfied": false,
  "draft_account_type_id": 1,
  "draft_routing_number": "111222333",
  "draft_account_number": "1234567890",
  "draft_account_name": "InvestorFirst InvestorLast",
  "status_id": 1,
  "account_type_id": 12,
  "w9_code_id": 1,
  "contact_name": "InvestorFirst InvestorLast",
  "contact_phone": "806-123-4567",
  "contact_email": "fi+rst.l+ast@somewhere.com",
  "idology_id": 123456,
  "created_by_ip_address": "123.456.789.012",
  "Assets": [],
  "Transactions": [
    {
      "id": 681354,
      "account_id": 81690,
      "asset_id": null,
      "date": "2015-01-02T00:00:00",
      "reference": "ref123",

```



```
    "description": "Cash Deposit",
    "amount": 5000,
    "status": "Processed",
    "created_by_ip_address": "123.456.789.012"
  },
  {
    "id": 6829651,
    "account_id": 81690,
    "asset_id": 471277,
    "date": "2015-01-02T00:00:00",
    "reference": "ref321",
    "description": "Purchase Assets",
    "amount": 250,
    "status": "Pending",
    "created_by_ip_address": "123.456.789.012"
  }
]
}
```

Update Account

Update Account – The only fields that are updatable are listed below. All other properties must be changed by communicating with GoldStar Trust. Please provide account_id (in red below) of account to update.

- mailing_address_1
- mailing_address_2
- mailing_city
- mailing_state
- mailing_zip
- mailing_country
- is_mailing_address_foreign
- portal_account_number
- contact_name,
- contact_phone
- contact_email

Request

Content-Type: application/json

PUT <https://qaservices.goldstartrust.com/Crowdfunding/api/Account/81700>

```
{
  "id": 81700,
  "number": "$1000604199",
  "portal_account_number": "youraccountnumber2",
  "investor_id": 76873,
  "name_1": "Changed Name",
  "name_2": null,
  "name_3": null,
  "name_4": null,
  "mailing_address_1": "123 Ave B",
  "mailing_address_2": null,
  "mailing_city": "Somewhere 2",
  "mailing_state": "TX",
  "mailing_zip": "79119",
  "mailing_country": null,
  "is_mailing_address_foreign": "false",
  "is_cip_satisfied": true,
  "draft_account_type_id": 1,
  "draft_routing_number": "111310870",
  "draft_account_number": "1234567892",
  "draft_account_name": "First Last 2",
  "status_id": 1,
  "account_type_id": 12,
  "w9_code_id": 1,
  "contact_name": "InvestorFirst InvestorLast 2",
  "contact_phone": "8061234562",
  "contact_email": "fi+rst.l+ast2@somewhere.com",
  "idology_id": 123452,
  "created_by_ip_address": "123.456.789.012"
}
```

Response

```
{
  "id": 81700,
  "number": "$1000604199",
  "portal_account_number": "youraccountnumber2",
  "investor_id": 76873,
  "name_1": "Changed Name",
  "name_2": null,
  "name_3": null,
  "name_4": null,
  "mailing_address_1": "123 Ave B",
  "mailing_address_2": null,
  "mailing_city": "Somewhere 2",
  "mailing_state": "TX",
  "mailing_zip": "79119",
  "mailing_country": null,
  "is_mailing_address_foreign": "false",
  "is_cip_satisfied": true,
  "draft_account_type_id": 1,
  "draft_routing_number": "111310870",
  "draft_account_number": "1234567892",
  "draft_account_name": "First Last 2",
  "status_id": 1,
  "account_type_id": 12,
  "w9_code_id": 1,
  "contact_name": "InvestorFirst InvestorLast 2",
  "contact_phone": "8061234562",
  "contact_email": "fi+rst.1+ast2@somewhere.com",
  "idology_id": 123452,
  "created_by_ip_address": "123.456.789.012",
  "Assets": [],
  "Transactions": []
}
```

Asset

Get Asset

Get information on asset specified by the `account_id` and `asset_id` parameters. The “All” keyword will return the asset using the `asset_id` searching at the `api key` level. The original GET continues to return the asset using the `asset_id` searching at the **`api key`** and **`portal key`** level.

Request

GET https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Assets/asset_id

GET https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Assets/asset_id/All

Response

```
{
  "id": 471277,
  "description": "ACME Wealth Agriculture 101, LLC",
  "number": "2000001",
  "sold_date": null,
  "market_value": 250,
  "Transactions": []
}
```

Get Asset (Version 2)

Same as Original version except that Asset model will be the version 2 model for Asset.

Request

GET https://qaservices.goldstartrust.com/Crowdfunding/api/V2/Account/account_id/Assets/asset_id

GET

https://qaservices.goldstartrust.com/Crowdfunding/api/V2/Account/account_id/Assets/asset_id/All

Response

```
{
  "id": 471277,
  "description": "ACME Wealth Agriculture 101, LLC",
  "number": "2000001",
  "sold_date": null,
  "market_value": 250,
  "term": null,
  "effective_date": "2015-04-15T00:00:00",
  "interest_type": "Simple",
  "interest_frequency": "Monthly",
  "interest_rate": 3.25,
  "maturity_date": "2017-10-15T00:00:00",
  "third_party_asset_number": "yourassetnumber",
  "quantity": 1,
  "Transactions": []
}
```

Get Asset with Transactions

Get information on asset specified by the `account_id` and `asset_id` parameters along with all transactions for the asset. The “All” keyword will return the asset and associated transactions using the `asset_id` searching at the `api` key level. The original GET continues to return the asset and associated transactions using the `asset_id` searching at the **`api key`** and **`portal key`** level.

Request

GET

`https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Assets/asset_id/transactions`

`https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Assets/asset_id/transactions/All`

Response

```
{
  "id": 471277,
  "description": "ACME Wealth Agriculture 101, LLC",
  "number": "2000001",
  "sold_date": null,
  "market_value": 250,
  "Transactions": [
    {
      "id": 6829651,
      "account_id": 81690,
      "asset_id": 471277,
      "date": "2015-01-02T00:00:00",
      "reference": "ref321",
      "description": "Purchase Assets",
      "amount": 250,
      "status": "Pending",
      "created_by_ip_address": "123.456.789.012"
    }
  ]
}
```

Get Asset with Transactions (Version 2)

Same as Original version except that Asset model will be the version 2 model for Asset.

Request

GET

https://qaservices.goldstarttrust.com/Crowdfunding/api/V2/Account/account_id/Assets/asset_id/transactions

https://qaservices.goldstarttrust.com/Crowdfunding/api/V2/Account/account_id/Assets/asset_id**/transactions/All**

Response

```
{
  "id": 471277,
  "description": "ACME Wealth Agriculture 101, LLC",
  "number": "2000001",
  "sold_date": null,
  "market_value": 250,
  "term": null,
  "effective_date": "2015-04-15T00:00:00",
  "interest_type": "Simple",
  "interest_frequency": "Monthly",
  "interest_rate": 3.25,
  "maturity_date": "2017-10-15T00:00:00",
  "third_party_asset_number": "yourassetnumber",
  "quantity": 1,
  "Transactions": [
    {
      "id": 6829651,
      "account_id": 81690,
      "asset_id": 471277,
      "date": "2015-01-02T00:00:00",
      "reference": "ref321",
      "description": "Purchase Assets",
      "amount": 250,
      "status": "Pending",
      "created_by_ip_address": "123.456.789.012"
    }
  ]
}
```

Escrows

Get Escrows

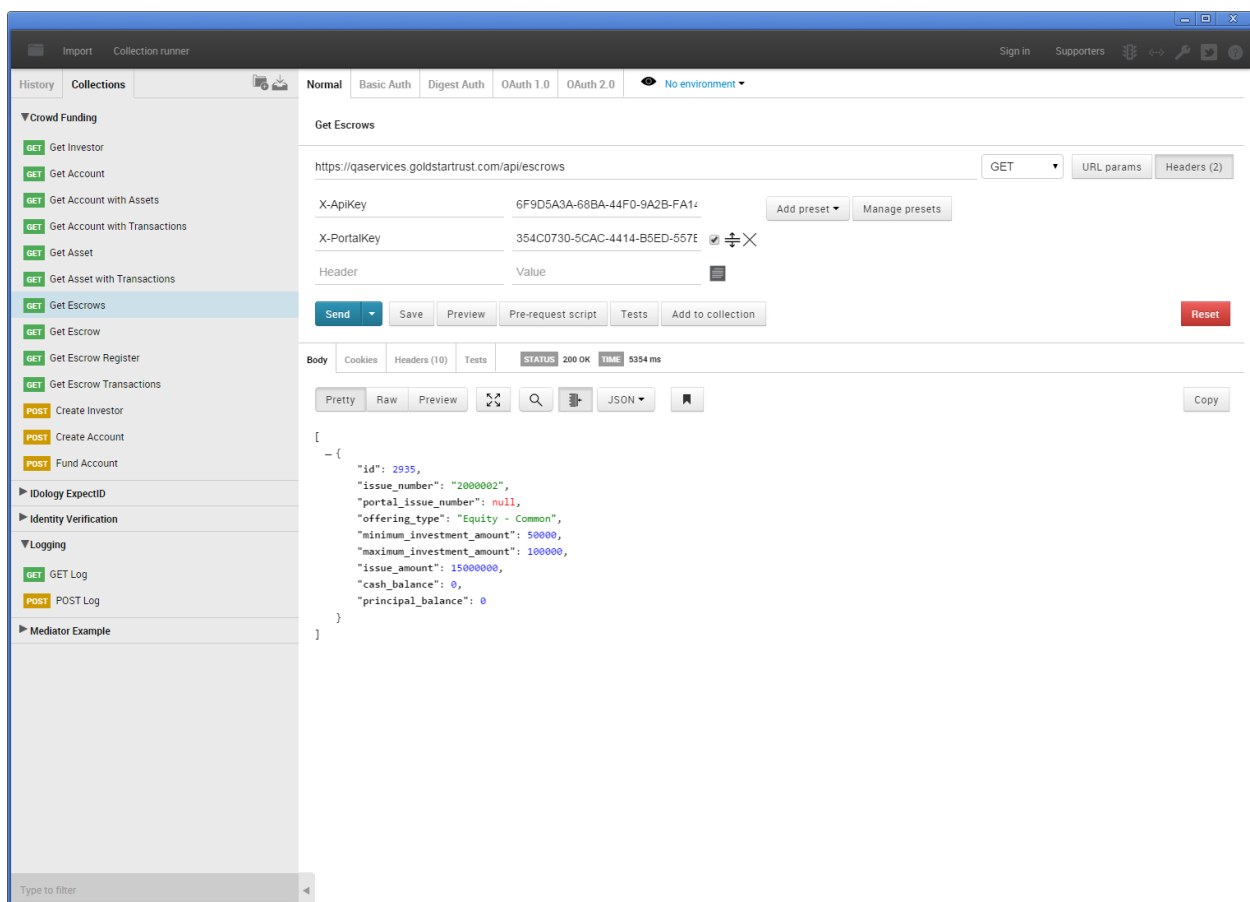
Returns a list of escrows associated with the X-PortalKey header value.

Request

GET <https://qaservices.goldstartrust.com/crowdfunding/api/Escrows>

Response

```
[
  {
    "id": 2935,
    "issue_number": "2000002",
    "portal_issue_number": null,
    "offering_type": "Equity - Common",
    "minimum_investment_amount": 50000,
    "maximum_investment_amount": 100000,
    "issue_amount": 15000000,
    "cash_balance": 0,
    "principal_balance": 0
  }
]
```



Get Escrow

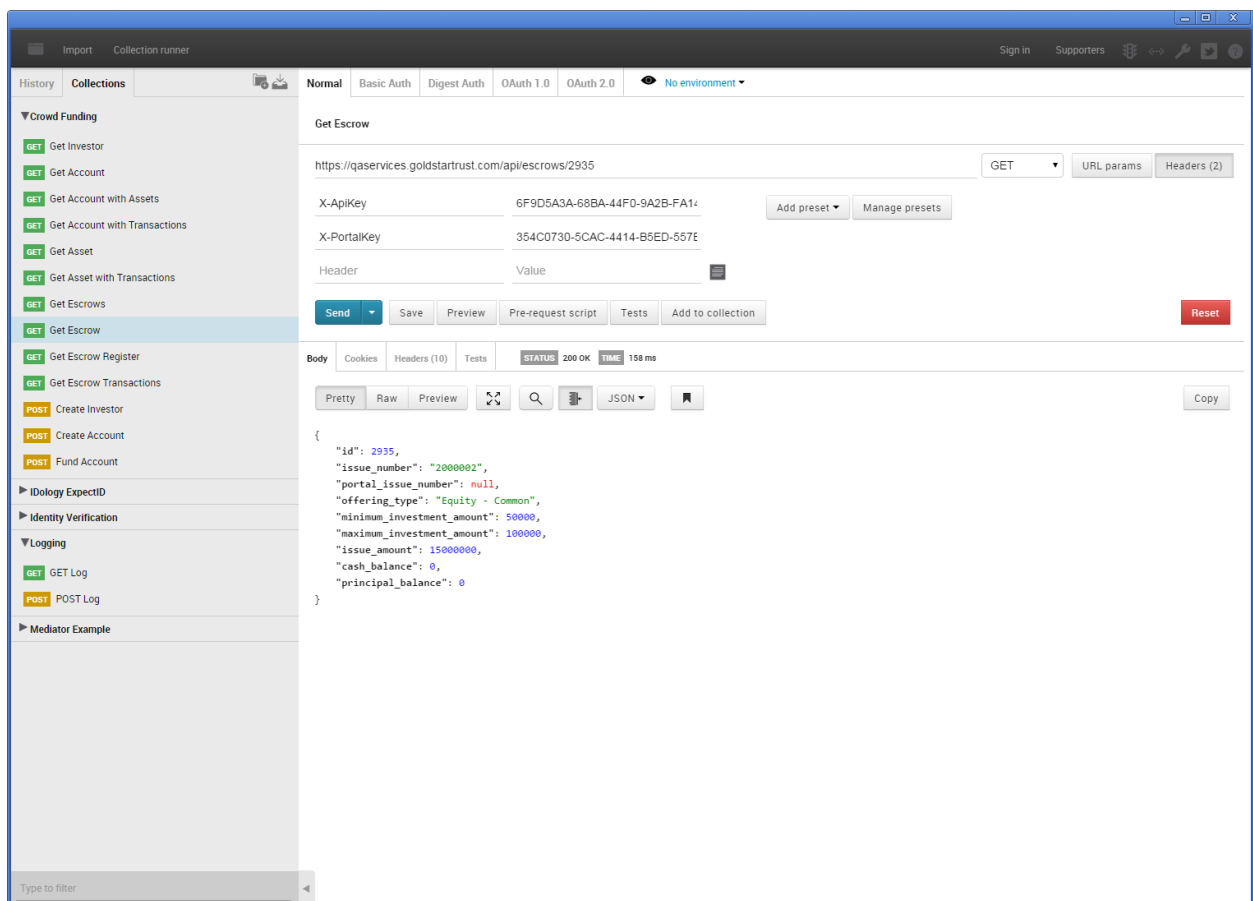
Returns the escrow by the given id parameter.

Request

GET <https://qaservices.goldstartrust.com/crowdfunding/api/Escrows/id>

Response

```
{
  "id": 2935,
  "issue_number": "2000002",
  "portal_issue_number": null,
  "offering_type": "Equity - Common",
  "minimum_investment_amount": 50000,
  "maximum_investment_amount": 100000,
  "issue_amount": 15000000,
  "cash_balance": 0,
  "principal_balance": 0,
  "Register": [ ],
  "Transactions": [ ]
}
```



Get Register

Returns a list of registrations for the given escrow id parameter.

Request

GET <https://qaservices.goldstartrust.com/crowdfunding/api/escrows/id/Register>

Response

```
{
  "id": 2935,
  "issue_number": "2000002",
  "portal_issue_number": null,
  "offering_type": "Equity - Common",
  "minimum_investment_amount": 50000,
  "maximum_investment_amount": 100000,
  "issue_amount": 15000000,
  "cash_balance": 0,
  "principal_balance": 0,
  "Register":
  [
    {
      "account_id": 81925,
      "asset_id": 471372,
      "asset_number": "2000002",
      "date": "2015-03-03T00:00:00",
      "quantity": 1,
      "amount": 2500,
      "status": "Processed"
    },
    {
      "account_id": 81700,
      "asset_id": 471374,
      "asset_number": "2000002",
      "date": "2015-03-03T00:00:00",
      "quantity": 1,
      "amount": 500,
      "status": "Pending"
    },
    {
      "account_id": 81700,
      "asset_id": 471401,
      "asset_number": "2000002",
      "date": "2015-03-07T00:00:00",
      "quantity": 1,
      "amount": 500,
      "status": "Pending"
    }
  ],
  "Transactions": [ ]
}
```

ImportCollection runner

Sign inSupporters

HistoryCollections

▼Crowd Funding

GET

Get Investor

GET

Get Account

GET

Get Account with Assets

GET

Get Account with Transactions

GET

Get Asset

GET

Get Asset with Transactions

GET

Get Escrows

GET

Get Escrow

GET

Get Escrow Register

GET

Get Escrow Transactions

POST

Create Investor

POST

Create Account

POST

Fund Account

►IDology ExpectID

►Identity Verification

►Logging

►Mediator Example

NormalBasic AuthDigest AuthOAuth 1.0OAuth 2.0No environment

Get Escrow Register

https://qaservices.goldstarttrust.com/api/escrows/2935/registerGETURL paramsHeaders (2)

X-APIKey6F9D5A3A-68BA-44F0-9A2B-FA14Add presetManage presets

X-PortalKey354C0730-5CAC-4414-B5ED-557E

HeaderValue

SendSavePreviewPre-request scriptTestsAdd to collectionReset

BodyCookiesHeaders (10)TestsSTATUS 200 OKTIME 149 ms

PrettyRawPreviewJSONCopy

```
{  "id": 2935,  "issue_number": "2000002",  "portal_issue_number": null,  "offering_type": "Equity - Common",  "minimum_investment_amount": 50000,  "maximum_investment_amount": 100000,  "issue_amount": 15000000,  "cash_balance": 0,  "principal_balance": 0,  - "Register": [    - {      "account_id": 77470,      "asset_id": 471271,      "asset_number": "2000002",      "date": "2014-12-02T00:00:00",      "quantity": 2,      "amount": 20000,      "status": "Processed"    },    - {      "account_id": 81840,      "asset_id": 471323,      "asset_number": "2000002",      "date": "2015-02-26T00:00:00",      "quantity": 1,      "amount": 200,      "status": "Processed"    }  ]}
```

Get Transactions

Returns a list of funding transactions or proceeds for the given escrow id parameter.

Request

GET <https://qaservices.goldstartrust.com/crowdfunding/api/Escrows/id/Transactions>

Response

```
{
  "id": 2935,
  "issue_number": "2000002",
  "portal_issue_number": null,
  "offering_type": "Equity - Common",
  "minimum_investment_amount": 50000,
  "maximum_investment_amount": 100000,
  "issue_amount": 15000000,
  "cash_balance": 0,
  "principal_balance": 0,
  "Register": [ ],
  "Transactions":
  [
    {
      "date": "2015-02-19T00:00:00",
      "description": "Purchase preferred stock",
      "amount": 10000,
      "status": "Processed"
    },
    {
      "date": "2015-02-20T00:00:00",
      "description": "Purchase preferred stock",
      "amount": 10000,
      "status": "Processed"
    },
    {
      "date": "2015-02-21T00:00:00",
      "description": "Purchase preferred stock",
      "amount": 10000,
      "status": "Pending"
    }
  ]
}
```

Transaction

Get Transaction

Returns the transaction for the given the **account_id** and **transaction_id** parameter. The “All” keyword will return the transaction using the transaction_id searching at the api key level. The original GET continues to return the transaction using the transaction_id searching at the **api key** and **portal key** level.

Request

GET

https://qaservices.goldstarttrust.com/crowdfunding/api/Account/account_id/Transaction/transaction_id

Response

```
{
  "amount": 100,
  "id": 6550796,
  "account_id": 82901,
  "asset_id": 471825,
  "date": "2015-04-07T00:00:00",
  "reference": "998877",
  "description": "funding escrow",
  "status": "Processed",
  "quantity": 0,
  "created_by_ip_address": "123.456.789.123"
}
```

Fund Account

Create ACH transaction to fund the Investor's crowd pay account. **account_id** parameter must be provided to associate transaction with the correct Crowd Pay account. Notice the **id** and **date** attributes returned in [Response](#). The **id** attribute is GoldStar Trust's internal unique identifier for a transaction. The **date** attribute is the date that the funds will become available in the crowd pay account. There will be a cutoff time for determining current date. The date attribute will be "plus 3" working days from the current date. If the transaction is submitted past 3:00 p.m. C.S.T., the current date will be the next date. Please see table below for examples.

Submission	Current Date	Transaction Date	Example	Transaction status	Account available balance
01/05/2015 2:30 p.m.	01/05/2015	01/08/2015	Before 3:00 p.m.	Pending – changes to Processed on transaction date	0.00 – added to transaction amount on transaction date
01/05/2015 3:30 p.m.	01/06/2015	01/09/2015	After 3:00 p.m.	Pending – changes to Processed on transaction date	0.00 – added to transaction amount on transaction date
01/09/2015 2:30 p.m.	01/09/2015	01/14/2015	Weekend Before 3:00 p.m.	Pending – changes to Processed on transaction date	0.00 – added to transaction amount on transaction date
01/09/2015 3:30 p.m.	01/10/2015	01/14/2015	Weekend After 3:00 p.m.	Pending – changes to Processed on transaction date	0.00 – added to transaction amount on transaction date
11/25/2015 2:30 p.m.	11/25/2015	12/01/2015	Day before bank holiday	Pending – changes to Processed on transaction date	0.00 – added to transaction amount on transaction date

Request

Content-Type: application/json

POST https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction/FundAccount

```
{
  "account_id": 1234567,
  "date": null,
  "reference": "yourreferencenumber",
  "description": "description you would like placed on transaction",
  "amount": 100.00,
  "created_by_ip_address": "123.456.789.012"
}
```

Response

Notice the `id` attribute returned. This attribute is Goldstar Trust's internal unique identifier for the transaction.

```
{
  "id": 9876543,
  "account_id": 1234567,
  "date": "2015-01-01T00:00:00",
  "reference": "yourreferencenumber",
  "description": "description you would like placed on transaction",
  "amount": 100.00,
  "status": "Pending",
  "created_by_ip_address": "123.456.789.012"
}
```

Fund Escrow

Initiate a crowd pay transaction to move funds from the investor account to the selected escrow account.

Please populate the `number_of_shares` property with 0 or null. The system will calculate the quantity.

Request

Content-Type: application/json

POST https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction/FundEscrow

```
{
  "account_id": 1234567,
  "date": null,
  "reference": "yourtransactionreferencenumber",
  "description": "description you would like placed on transaction",
  "amount": 100.00,
  "quantity": 10,
  "created_by_ip_address": "123.456.789.012"
}
```

Response

Notice the `id` attribute returned. This attribute is Goldstar Trust's internal unique identifier for the transaction.

```
{
  "id": 9876544,
  "account_id": 1234567,
  "asset_id": 1234567,
  "date": "2015-01-06T00:00:00",
  "reference": "yourreferencenumber",
  "description": "description you would like placed on transaction",
  "amount": 100.00,
  "quantity": 10,
  "created_by_ip_address": "123.456.789.012"
}
```

Fund Debt Escrow

Initiate a crowd pay transaction to move funds from the investor account to the selected escrow account with additional debt asset properties.

Submission	Current Date	Transaction Date	Example	Transaction status	Account available balance
01/05/2015 2:30 p.m.	01/05/2015	01/05/2015	Before 3:00 p.m.	Pending Changes to Processed when GoldStar "Posts" transactions on 01/05/2015	Transaction amount subtracted from available balance after GoldStar "Posts" transactions on 01/05/2015
01/05/2015 3:30 p.m.	01/06/2015	01/06/2015	After 3:00 p.m.	Pending Changes to Processed when GoldStar "Posts" transactions on 01/06/2015	Transaction amount subtracted from available balance after GoldStar "Posts" transactions on 01/06/2015

Request

Content-Type: application/json

POST https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction/FundDebtEscrow

```
{
  "account_id": 1234567,
  "date": null,
  "reference": "yourtransactionreferencenumber",
  "description": "description you would like placed on transaction",
  "amount": 100.00,
  "quantity": 1,
  "created_by_ip_address": "123.456.789.012"
  "term": 3.0,
  "effective_date": "2015-01-15T00:00:00",
  "interest_type": 1,
  "interest_frequency": 3,
  "interest_rate": 1.50,
  "maturity_date": "2015-07-01T00:00:00",
  "third_party_asset_number": "yourassetnumber",
  "asset_description": "custom asset description",
  "cusip_number": "1315020180115"
```



```
}
```

Response

Notice the `id` and `asset_id` attribute returned. This `id` is GoldStar Trust's internal unique identifier for the transaction. The `asset_id` is GoldStar Trust's internal unique identifier for the asset created. The `asset_id` can be used with Get Asset and Get Asset with Transactions api functions.

```
{
  "id": 9876544,
  "account_id": 1234567,
  "asset_id": 234567,
  "date": "2015-01-06T00:00:00",
  "reference": "yourreferencenumber",
  "description": "description you would like placed on transaction",
  "amount": 100.00,
  "quantity": 1,
  "created_by_ip_address": "123.456.789.012",
  "term": 10.5,
  "effective_date": "2015-01-15T00:00:00",
  "interest_type": 1,
  "interest_frequency": 1,
  "interest_rate": 3.250,
  "maturity_date": "2015-07-01T00:00:00",
  "third_party_asset_number": "yourassetnumber",
  "asset_description": "custom asset description"
  "cusip_number": "1315020180115"
}
```

Debt Reinvest

Initiate a crowd pay transaction to mature a debt asset and reinvest partial or total amount of the funds into another debt asset.

Submission	Current Date	Transaction Date	Example	Transaction status	Account available balance
01/05/2015 2:30 p.m.	01/05/2015	01/05/2015	Before 3:00 p.m.	Pending Changes to Processed when GoldStar "Posts" transactions on 01/05/2015	Transaction amount subtracted from available balance after GoldStar "Posts" transactions on 01/05/2015
01/05/2015 3:30 p.m.	01/06/2015	01/06/2015	After 3:00 p.m.	Pending Changes to Processed when GoldStar "Posts" transactions on 01/06/2015	Transaction amount subtracted from available balance after GoldStar "Posts" transactions on 01/06/2015

Request

Content-Type: application/json

POST https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction/ReinvestDebt

******* Very Important *******

asset_id in content must represent the asset_id of the debt asset that you wish to mature.

sold_date – if null, we will use transaction date (**you should leave as null unless you know what you want valued in this field**)

description – we will provide default description and return in response if you leave blank

activity_code

- **707** – Total Reinvestment
 - Enter amount = 0 or book value/cost basis
- **708** – Partial Reinvestment
 - Enter amount = to amount that you wish to reinvest
 - The difference in amount entered and book value/cost basis will show up as cash in crowd pay account

{

```

"account_id": 1234567,
"asset_id": 987654,
"activity_code": "707",
"sold_date": null,
"reference": "yourtransactionreferencenumber",
"description": "description you would like placed on transaction",
"amount": 100.00,
"quantity": 1,
"created_by_ip_address": "123.456.789.012"
"term": 3.0,
"effective_date": "2015-01-15T00:00:00",
"interest_type": 1,
"interest_frequency": 3,
"interest_rate": 1.50,
"maturity_date": "2018-07-15T00:00:00",
"third_party_asset_number": "yourassetnumber",
"asset_description": "custom asset description",
"cusip_number": "1315020180115"
}

```

Response

Notice the **id** and **asset_id** attribute returned. This **id** is GoldStar Trust's internal unique identifier for the transaction. The **asset_id** is GoldStar Trust's internal unique identifier for the new asset created. The **asset_id** can be used with Get Asset and Get Asset with Transactions api functions.

```

{
  "id": "1223344"
  "account_id": 1234567,
  "asset_id": 999888,
  "activity_code": "707",
  "sold_date": null,
  "reference": "yourtransactionreferencenumber",
  "description": "description you would like placed on transaction",
  "amount": 100.00,
  "quantity": 1,
  "created_by_ip_address": "123.456.789.012"
  "term": 3.0,
  "effective_date": "2015-01-15T00:00:00",
  "interest_type": 1,
  "interest_frequency": 3,
  "interest_rate": 1.50,
  "maturity_date": "2018-07-15T00:00:00",
  "third_party_asset_number": "yourassetnumber",
  "asset_description": "custom asset description",
  "cusip_number": "1315020180115",
  "status": "pending"
}

```

Debt Pay

Initiate a crowd pay transaction to post interest or principal to a debt asset. In addition, this api function can also mature an asset and apply the cash to the crowd pay account of the matured asset.

Submission	Current Date	Transaction Date	Example	Transaction status	Account available balance
01/05/2015 2:30 p.m.	01/05/2015	01/05/2015	Before 3:00 p.m.	Pending Changes to Processed when GoldStar "Posts" transactions on 01/05/2015	Transaction amount subtracted from available balance after GoldStar "Posts" transactions on 01/05/2015
01/05/2015 3:30 p.m.	01/06/2015	01/06/2015	After 3:00 p.m.	Pending Changes to Processed when GoldStar "Posts" transactions on 01/06/2015	Transaction amount subtracted from available balance after GoldStar "Posts" transactions on 01/06/2015

Request

Content-Type: application/json

POST https://qaservices.goldstartrust.com/Crowdfunding/api/Account/account_id/Transaction/DebtPay

******* Very Important *******

asset_id in content must represent the asset_id of the debt asset that you wish to mature.

activity_code

- **301** – amount = amount of interest income that you wish to apply to asset
- **304** – amount = amount of increase in asset principal
- **305** – Sell/mature asset
 - Enter **amount** = 0 or book value/cost basis – The asset's book value will show up as cash in crowd pay account
 - **sold_date** – if null, we will use transaction date (**you should leave as null unless you know what you want valued in this field**)

```
{  
  "account_id": 1234567,  
  "asset_id": 987654,  
  "activity_code": "305",  
}
```

```

    "sold_date": null,
    "reference": "yourtransactionreferencenumber",
    "description": "description you would like placed on transaction",
    "amount": 100.00,
    "created_by_ip_address": "123.456.789.012"
}

```

Response

Notice the **id**. This **id** is GoldStar Trust's internal unique identifier for the transaction. The `asset_id` can be used with Get Asset and Get Asset with Transactions api functions.

```

{
  "id": 6580369,
  "account_id": 1234567,
  "asset_id": 987654,
  "activity_code": "305",
  "sold_date": 2015-06-01T00:00:00,
  "reference": "yourtransactionreferencenumber",
  "description": "description you would like placed on transaction",
  "amount": 100.00,
  "created_by_ip_address": "123.456.789.012",
  "status": "pending"
}

```