

MMRRC ISCS API

The MMRRC ICSC API is a tool for the general public as well as MMRRC centers to use for programmatically interacting with the MMRRC catalog. The API will contain functions to query the catalog for individual products, the entire catalog, or to submit and edit catalog entries from MMRRC and affiliated centers.

One of the primary uses for the API is to allow KOMP production centers to programmatically submit their production data directly into the MMRRC and speed up the curation and approval process of new strains produced as part of the KOMP2 Phase 2 effort.

API Framework Details: This RESTful API was built using the PHP micro MVC framework [SILE X](#) and interacts by producing and consuming objects in the JSON lightweight data interchange format. Examples of the specific functions, their usage, and the JSON formats produced and expected are detailed below.

Table of Contents

- [MMRRC ISCS API](#)
 - [Version](#)
 - [Change history](#)
 - [Functions and usage](#)
 - [Sandbox access](#)
 - [Conventions](#)
 - [Endpoints](#)
 - [List of Endpoints](#)
 - [Authorization](#)
 - [Error reporting](#)
 - [List of GET functions](#)
 - [schema](#)
 - [getCatalog\(page_number\)](#)
 - [getCatalogEntry\(mmrrc_id\)](#)
 - [getStrain\(mmrrc_id\)](#)
 - [getSDS\(mmrrc_id\)](#)
 - [getLive\(center\)](#)
 - [getAllOrders\(\)](#)

- [getActiveOrders\(\)](#)
- [getOrderDetails\(order_id\)](#)
- List of PUT functions
 - [updateOrderStatus\(\)](#)
- List of POST functions
 - [addOrderNote\(\)](#)
 - [addOrderMilestone\(\)](#)
 - [addOrderShipment\(\)](#)
- [Creating a new submission](#)
- [Updating an existing submission](#)
- [Caveats:](#)
- [Useful Tools:](#)
 - [HTML Response](#)
 - [JSON linter](#)
 - [cURL](#)

Version

Version 1.2

Change history

Version 1.2 2019-03-08 - added documentation for updating submissions
Version 1.1 2017-04-28

Functions and usage

Sandbox access

We have made a sandbox available for developers to test their implementation of this API at api.sandbox.mmrrc.org.

Conventions

Text annotations for strain names are stored as Unicode characters.

Endpoints

Endpoints are categorized into three groups based on the HTTP verb used to make a successful request: GET, POST and PUT.

GET endpoint functions are used for data retrieval.

POST endpoint is used to create a new submission.

PUT endpoint will be used to edit a submission already existing in the system.

GET endpoint functions are public and do not require authorization. POST and PUT requests are private and accessible only on successful authorization. Refer to 'Authorization' for more information.

List of Endpoints

URI	HTTP verb	Description
/api/v1/schema	GET	See the submission schema
/api/v1/catalog/{function}/{argument}	GET	JSON output of all catalog items /submissions
/api/v1/submission/	POST	Create a new submission
/api/v1/submission/updateSubmission/{mmrrc_id}	PUT	Edit fields for a specific submission
/api/v1/orders{function}/{order_id}	GET	JSON output of a specified order or all orders
/api/v1/orders/{function}	POST, PUT	Add/Edit order information

Authorization

POST and PUT requests are authorized via the x-api-key HTTP header. Every such request should contain an x-api-key HTTP header with a unique API key as its value. API keys are associated with individual API users (centers) and can be obtained from service@mmrrc.org.

To quickly check if an API key is valid, the following cURL command can be used.

```
$ curl --header "x-api-key: eceab44fdb2d86fx" -X POST --data-binary
 '{"sample":"value"}' http://api.mmrrc.org/api/v1/submission
```

If the x-api-key value is incorrect or missing, the API returns the following error message:

```
{
  "failure": {
    "errors": [
      "Your 'x-api-key' header value is incorrect or empty."
    ]
  }
}
```

Error reporting

The API uses descriptive human-readable error messages for troubleshooting. An example errors object:

```
{
  "failure": {
    "errors": [
      "An entry with a strain name _C57BL/6J-A_13_ already exists."
    ]
  }
}
```

List of GET functions

schema

This function retrieves the schema for both the submission and update calls.

Endpoint: /api/v1/schema/

Example request (GET): <https://api.mmrrc.org/api/v1/schema>

Example response:

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Submission",
  "description": "MMRRC Submission",
  "type": "object",
  "definitions": {
    "contactType": {
      "properties": {
        "first_name": {
          "type": "string"
        },
        "last_name": {
          "type": "string"
        },
        "institution": {
          "type": "string"
        },
        "email_address": {
          "type": "string"
        },
        "phone": {
          "type": "string"
        }
      }
    }
  },
  "more": "...
}
```

Further down the schema enumerated values are presented as:

```
{
  "breeding_system": {
    "enum": [
      "Backcross",
      "Sib-mating",

```

```

    "Random intra-strain mating",
    "Outcross to non inbred (hybrid or outbred) mate",
    "Other or uncertain",
    "Backcross or Sib-Mating"
  ]
}
}

```

and required values for the submission call are denoted in an array at the end of each section:

```

{
  "section": "...",
  "required": [
    "strain_name",
    "status",
    "submission_type",
    "accepted_dt",
    "strain_type_id",
    "delayed_release_requested",
    "publication_provided",
    "originated_on",
    "maintained_on",
    "mta_rcvd_dt"
  ]
}

```

getCatalog(page_number)

This function retrieves the entire catalog in JSON format. The response is paginated, i.e. a single request will return 100 catalog entries starting at page_number.

Endpoint: /api/v1/catalog/getCatalog/

Example request (GET): <https://api.mmrrc.org/api/v1/catalog/getCatalog/400>

Example response:

```

[
  {
    "mmrrc_id": "41491",
    "strain_name": "B6;CBA-Tg(MYH6-Mb*)10Jue/Mmucd",
    "status": "Accepted",
    "accepted_dt": "2016-04-19",
  }
]

```

```

    "delayed_release_requested": "0",
    "published_name": "B6;CBA-TG(HpMyo)10Jue",
    "is_transgenic": "Yes",
    "founder_background": "B6CBAF1",
    "hom_fertile_comment": "",
    "is_sexlinked": "No",
    "genetic_alteration": "The transgene included anti-sense...",
    "phenotype": "<p><b>Homozygous: </b>Not characterized</p>...",
    "availability_level": "",
    "center_id": "UCD",
    "assigned_dt": "2016-04-19",
    "availability": "Cryo-archive",
    "locus": [
      {
        "mmrrc_id": "41491",
        "symbol": "Tg(MYH6-Mb*)10Jue",
        "segment_type": "Transgene",
        "species": "Multi-species",
        "locus_id": "45388",
        "mgi_acc_id": "",
        "locus_role": "Allele"
      }
    ],
    "more": "...
  }
]

```

getCatalogEntry(mmrrc_id)

This function returns a single catalog entry by mmrrc_id.

Endpoint: /api/v1/catalog/getCatalogEntry/

Example request (GET): <https://api.mmrrc.org/api/v1/catalog/getCatalogEntry/44>

Example response:

```

{
  "mmrrc_id": "44",
  "strain_name": "B6;129-AmelxtmlKul/Mmnc",
  "status": "Accepted",
  "accepted_dt": "2002-02-26",
  "delayed_release_requested": "0",
  "ilar_lab_code": "",
  "published_name": "",
  "is_transgenic": "No",
  "founder_background": "BL6;129SvJ",

```

```

"hom_fertile_comment": "",
"is_sexlinked": "No",
"genetic_alteration": "A neo cassette was inserted in exon 2...",
"phenotype": "Abnormal teeth with chaulky-white discoloration,...",
"availability_level": "",
"center_id": "UNC",
"assigned_dt": "2002-02-26",
"availability": "Cryo-archive",
"locus": [
  {
    "mmrrc_id": "44",
    "symbol": "Amelx",
    "segment_type": "Gene",
    "species": "Mus musculus (mouse)",
    "locus_id": "5",
    "mgi_acc_id": "MGI:88005",
    "locus_role": "Target"
  },
  {
    "mmrrc_id": "44",
    "symbol": "Amelx<tm1Kul>",
    "segment_type": "Allele",
    "species": "Mus musculus (mouse)",
    "locus_id": "6",
    "mgi_acc_id": "MGI:2177294",
    "locus_role": "Target"
  }
],
"strain_of_origin": "BL6;129SvJ",
"background_strain": "BL6;129SvJ",
"product": {
  "prod_hom_f": "1",
  "prod_hom_m": "1",
  "prod_het_hemi_f": "1",
  "prod_het_hemi_m": "1",
  "prod_sperm": "0",
  "prod_embryos": "0",
  "prod_resusc": "1",
  "prod_wt_control_f": "1",
  "prod_wt_control_m": "1",
  "prod_es_cell_line": "0"
}
}

```

getStrain(mmrrc_id)

This function retrieves strain information by mmrrc_id.

Endpoint: /api/v1/catalog/getStrain/

Example request (GET): <https://api.mmrrc.org/api/v1/catalog/getStrain/67>

Example response:

```
{
  "mmrrc_id": "67",
  "strain_name": "FVB\N-Tg(Myh7-Pln)2Egk\Mmmh",
  "nomen_sanctioned": "1",
  "mgi_acc_id": "MGI:2665588",
  "accepted_dt": "2002-06-18",
  "delayed_release_dt": "",
  "mta_version": "1",
  "mta_rcvd_dt": "2002-07-22",
  "mta_release_category": "B",
  "initial_maintenance_level": "Cryo-archive",
  "historical_dev_desc": "",
  "originated_on": "",
  "nomen_notes": "Changed name from \u0027FVB-Tg(Pln)2Egk\u0027 to...",
  "maintained_on": "",
  "is_di_protocol_onfile": "1",
  "is_profile_onfile": "1",
  "hide_related_products": "N",
  "other_notes": "Previously assigned to Taconic. Non-Profit Only...",
  "created_dt": "2002-06-18",
  "created_by": "MMRRCDBA",
  "mod_dt": "2016-03-30 12:25:06",
  "mod_by": "etpage@ucdavis.edu",
  "strain_type_id": "15",
  "submission_type": "Type-1",
  "status": "Accepted",
  "nomen_chng_dt": "",
  "avd_level": "Cryo-archive",
  "source_collection": "",
  "delayed_release_extns": "0",
  "delayed_release_requested": "0",
  "publication_provided": "0",
  "donor_requested_delay": "0",
  "create_dt": "",
  "create_by": "0"
}
```

getSDS(mmrrc_id)

This function retrieves all the data found on the Strain Detail Sheet for a single ID.

Endpoint: /api/v1/catalog/getSDS/

Example request (GET): <https://api.mmrrc.org/api/v1/catalog/getSDS/67>

Example response:

```
{
  "mmrrc_id": "67",
  "doc_title": "<title>MMRRC:000067-MU Detail Page</title>",
  "published_name": "",
  "genetic_alteration": "Transgene",
  "genotyping": "<ul><li><a href=\u0022http://www.mu-mmrrc.com/...>",
  "es_cell_line": "",
  "phenotype": "Overexpression of phospholamban was expressed in...",
  "founder_background": "FVB/N",
  "current_background": "FVB/N",
  "strain_development": "",
  "research_applications": "<li>Skeletal muscle contraction</li>...",
  "donor_institution": "Dr. Evangelia Kranias, University of...",
  "reference": "",
  "coat_color": "White",
  "eye_color": "",
  "other_appearance": "",
  "breeding_system": "Backcross / Sib mating",
  "breeding_schemes": "<ul><li>FVB/NTac female x Hemizyous Tg male...>",
  "generation": "Unknown",
  "breeding_performance": "Good",
  "reproductive_statistics": "<tr><td align=\u0022right\u0022>...>",
  "ave_pups_born": "8",
  "homozgosity_yn": "",
  "average_pups_weaned": "",
  "wean_age": "3 weeks",
  "special_husbandry": "",
  "available_control": "<ul><li> Wildtype littermates</li>...",
  "parent_cell_line": "",
  "background_strain": "",
  "culture_protocol": "",
  "health_status_cryo": "<a href=\u0022http://www.mu-mmrrc.com/...>",
  "availability_level": "",
  "hide_colony_husbandry_info": "No",
  "phenotype_list": "<div class=\u0022mmrrcpage\u0022> <br>...",
  "f_hom_viable": "",
  "m_hom_viable": "",
  "f_hom_fertile": "",
  "m_hom_fertile": "",
  "f_het_hemi_fertile": "",
  "m_het_hemi_fertile": "",
  "f_age_repdec": "",
  "m_age_repdec": "",
  "update_dt": "2019-02-17 04:58:05",
  "user_id": "6852",
```

```
"is_transgenic": "",
"is_sexlinked": "",
"public_display": "Yes",
"public_display_dt": "2019-02-17",
"show_repro_stat": "No",
"breeding_comment": "",
"sds_status": "Complete",
"hom_viable_comment": "",
"hom_fertile_comment": "",
"het_hemi_fertile_comment": "",
"protocol_status": "Needs Protocol",
"donor_notes": "",
"edit_notes": "",
"mammalian_phenotype_id": "",
"sbm_gene_symbol": "",
"sbm_allele_pairs": "",
"sbm_additional_allele_info": "",
"esc_please_see_also": "",
"esc_cell_development": "",
"created_dt": "",
"alterations": [
  {
    "locus_id": "97",
    "genetic_alteration_id": "61",
    "alteration_id": "6",
    "alteration_desc": "Transgene.",
    "mmrrc_id": "67",
    "cell_line_id": "902",
    "additional_notes": "",
    "presentation_order": "0",
    "alteration": "Transgenic",
    "is_standard": "1",
    "subtype_of": "",
    "imsr_alteration_cd": "TG",
    "note": "",
    "locus_role": "Transgene",
    "mgi_acc_id": "MGI:97622",
    "symbol": "Pln",
    "markup_symbol": "Pln",
    "name": "phospholamban",
    "chromosome": "10",
    "species": "Mus musculus (mouse)",
    "synonyms": "<sup>",
    "segment_type": "Gene",
    "component_of": "",
    "locus_notes": "",
    "external_acc_id": "MGI:97622",
    "resource_id": "MGI",
    "last_modified_dt": "2019-02-17 06:49:10",
    "created_dt": ""
  },
  {
```

```

"locus_id": "3946",
"genetic_alteration_id": "61",
"alteration_id": "6",
"alteration_desc": "Transgene.",
"mmrrc_id": "67",
"cell_line_id": "902",
"additional_notes": "",
"presentation_order": "0",
"alteration": "Transgenic",
"is_standard": "1",
"subtype_of": "",
"imsr_alteration_cd": "TG",
"note": "",
"locus_role": "Promoter",
"mgi_acc_id": "MGI:2155600",
"symbol": "Myh7",
"markup_symbol": "Myh7",
"name": "myosin, heavy polypeptide 7, cardiac muscle, beta",
"chromosome": "14",
"species": "Mus musculus (mouse)",
"synonyms": "<sup>",
"segment_type": "Gene",
"component_of": "",
"locus_notes": "",
"external_acc_id": "MGI:2155600",
"resource_id": "MGI",
"last_modified_dt": "2019-02-17 06:49:02",
"created_dt": ""
},
{
"locus_id": "35851",
"genetic_alteration_id": "61",
"alteration_id": "6",
"alteration_desc": "Transgene.",
"mmrrc_id": "67",
"cell_line_id": "902",
"additional_notes": "",
"presentation_order": "0",
"alteration": "Transgenic",
"is_standard": "1",
"subtype_of": "",
"imsr_alteration_cd": "TG",
"note": "",
"locus_role": "Transgene",
"mgi_acc_id": "MGI:3834218",
"symbol": "Tg(Myh7-Pln)2Egk",
"markup_symbol": "Tg(Myh7-Pln)2Egk",
"name": "transgene insertion 2, Evangelia G Kranias;...",
"chromosome": "unknown",
"species": "Mus musculus (mouse)",
"synonyms": "Pln<OE>|Tg(Myh6-Pln)2Egk|Tg(Pln)2Egk",
"segment_type": "Transgene",

```

```

        "component_of": "35853",
        "locus_notes": "",
        "external_acc_id": "MGI:3834218",
        "resource_id": "MGI",
        "last_modified_dt": "2019-02-17 06:36:36",
        "created_dt": ""
    }
]
}

```

getLive(center)

This function retrieves a list of live mice available from the specified center.

Endpoint: /api/v1/catalog/getLive/

Example request (GET):

<https://api.mmrrc.org/api/v1/catalog/getLive/UCD>

Example response:

```

[
  {
    "mmrrc_id": "30648",
    "link": "<a href=\u0022mmrrc.org/catalog/sds.php?mmrrc_id=30648\u0022>B6.FV",
    "normalized_strain_name": "B6.FVB(Cg)-Tg(Ntsr1-cre)GN220Gsat/Mmucd",
    "rrid": "",
    "source_collection": "GENSAT",
    "withdrawn_dt": "",
    "mutations": [
      {
        "text": "<b><a target=\u0022_blank\u0022 href=\u0022http://www.info",
      },
      {
        "text": "<b><a target=\u0022_blank\u0022 href=\u0022http://www.ncbi",
      },
      {
        "text": "<b><a target=\u0022_blank\u0022 href=\u0022http://www.info",
        "alteration": "Transgenic (cre)"
      }
    ]
  }
]

```

getAllOrders()

This function retrieves all orders for the center associated with an API user making this request. It is only possible to retrieve orders that were issued by the center that is associated with the API key being used.

Endpoint: /api/v1/catalog/getAllOrders/

Example request (GET): <https://api.mmrrc.org/api/v1/catalog/getAllOrders>

Example response:

```
[
  {
    "mmrrc_order_oid": "7845472",
    "actual_ship_dt": "2007-03-05",
    "billing_acct": "",
    "billing_id": "",
    "billing_subacct": "",
    "center_id": "UCD",
    "center_order_id": "",
    "center_ping_dt": "",
    "completed_by_email": "",
    "completed_by_name": "",
    "completed_by_title": "",
    "estd_ship_dt": "",
    "fedex_acct_num_tx": "",
    "init_estd_ship_dt": "",
    "instructions_comments_tx": "2\16\07 request date",
    "is_for_nih_grant": "0",
    "is_hlth_stat_preship_reqd": "0",
    "...": "more"
  }
]
```

getActiveOrders()

This function retrieves all active orders for the center associated with an API user making this request. It is only possible to retrieve orders that were issued by the center that is associated with the API key being used.

Endpoint: /api/v1/catalog/getActiveOrders/

Example request (GET): <https://api.mmrrc.org/api/v1/catalog/getActiveOrders>

Example response:

```
see the response for the getAllOrders() call
```

getOrderDetails(order_id)

This function retrieves order information by order_id. The requester can only access orders of their center.

Endpoint: /api/v1/catalog/getOrderDetails/

Example request (GET): <https://api.mmrrc.org/api/v1/catalog/getOrderDetails/7827472>

Example response:

```
{
  "order_info": {
    "mmrrc_order_oid": "7827472",
    "actual_ship_dt": "2007-03-05",
    "billing_acct": "",
    "billing_id": "",
    "billing_subacct": "",
    "center_id": "UCD",
    "center_order_id": "",
    "center_ping_dt": "",
    "completed_by_email": "",
    "completed_by_name": "",
    "completed_by_title": "",
    "estd_ship_dt": "",
    "fedex_acct_num_tx": "",
    "init_estd_ship_dt": "",
    "instructions_comments_tx": "2/16/07 request date",
    "is_for_nih_grant": "0",
    "is_hlth_stat_preship_reqd": "0",
    "is_paying_credit_card": "0",
    "is_ri_planning_make_mice": "1",
    "po_expiration_dt": "",
    "po_num_tx": "726135",
    "proj_code": "",
    "icc_center_oid": "",
    "order_submission_oid": "7827461",
```

```
"state_id": "3206",
"lockedbysession_id": "",
"create_by": "6879",
"create_dt": "2007-06-06",
"mod_by": "267447",
"mod_dt": "2010-10-17",
"order_dt": "2007-02-16",
"preferred_carrier_acct": "",
"preferred_carrier_tx": "",
"customer_class_cd": "NP",
"preferred_pay_method": "PO",
"completed_by_phone": "",
"ip_responsible_role_tx": "Bill-to",
"other_emails": "",
"customer_label": "",
"grant_numbers": "",
"center_accept_dt": "",
"waive_quarantine": "",
"nih_sources": "",
"state_cd": "Fulfilled"
},
"contacts": [{
  "role_cd": "Bill-to",
  "order_contact_oid": "7847476",
  "distribution_contact_oid": "28429588",
  "brim_user_id": "",
  "email_address": "caroline@example.com",
  "fax": "254-618-1420",
  "first_name": "Caroline ",
  "is_active": "1",
  "last_name": "Smith",
  "name_suffix": "",
  "title": "",
  "voice_phone": "456-634-1134",
  "create_by": "267447",
  "create_dt": "2010-10-17",
  "mod_by": "",
  "mod_dt": "",
  "distrib_cont_instn_oid": "28449589",
  "institution_oid": "28449587",
  "state_province": "TX",
  "postal_code": "76235-5303",
  "country": "United States",
  "city": "Dallas ",
  "address": "6000 Oak Street",
  "address_detail": "",
  "institution_name": "University of Texas"
}, {
  "role_cd": "Recipient",
  "order_contact_oid": "7847477",
  "distribution_contact_oid": "28429588",
  "brim_user_id": "",
```

```
"email_address": "caroline@example.com",
"fax": "254-618-1420",
"first_name": "Caroline ",
"is_active": "1",
"last_name": "Smith",
"name_suffix": "",
"title": "",
"voice_phone": "456-634-1134",
"create_by": "267447",
"create_dt": "2010-10-17",
"mod_by": "",
"mod_dt": "",
"distrib_cont_instn_oid": "28449589",
"institution_oid": "28449587",
"state_province": "TX",
"postal_code": "76235-5303",
"country": "United States",
"city": "Dallas ",
"address": "6000 Oak Street",
"address_detail": "",
"institution_name": "University of Texas"
}, {
  "role_cd": "Ship-to",
  "order_contact_oid": "7827478",
  "distribution_contact_oid": "",
  "brim_user_id": "",
  "email_address": "",
  "fax": "",
  "first_name": "",
  "is_active": "",
  "last_name": "",
  "name_suffix": "",
  "title": "",
  "voice_phone": "",
  "create_by": "",
  "create_dt": "",
  "mod_by": "",
  "mod_dt": "",
  "distrib_cont_instn_oid": "",
  "institution_oid": "",
  "state_province": "TX",
  "postal_code": "76235-5303",
  "country": "United States",
  "city": "Dallas",
  "address": "6000 Oak Street",
  "address_detail": "",
  "institution_name": ""
}, {
  "role_cd": "Ship-to Contact",
  "order_contact_oid": "7847477",
  "distribution_contact_oid": "28429588",
  "brim_user_id": "",
```

```
"email_address": "caroline@example.com",
"fax": "254-618-1420",
"first_name": "Caroline ",
"is_active": "1",
"last_name": "Smith",
"name_suffix": "",
"title": "",
"voice_phone": "456-634-1134",
"create_by": "267447",
"create_dt": "2010-10-17",
"mod_by": "",
"mod_dt": "",
"distrib_cont_instn_oid": "28449589",
"institution_oid": "28449587",
"state_province": "TX",
"postal_code": "76235-5303",
"country": "United States",
"city": "Dallas ",
"address": "6000 Oak Street",
"address_detail": "",
"institution_name": "University of Texas"
}, {
  "role_cd": "Vet",
  "order_contact_oid": "7827482",
  "distribution_contact_oid": "",
  "brim_user_id": "",
  "email_address": "",
  "fax": "",
  "first_name": "",
  "is_active": "",
  "last_name": "",
  "name_suffix": "",
  "title": "",
  "voice_phone": "",
  "create_by": "",
  "create_dt": "",
  "mod_by": "",
  "mod_dt": "",
  "distrib_cont_instn_oid": "",
  "institution_oid": "",
  "state_province": "",
  "postal_code": "",
  "country": "",
  "city": "",
  "address": "",
  "address_detail": "",
  "institution_name": ""
}
],
"restrictions": [{
  "order_strain_restriction_oid": "7827487",
  "closed_dt": "2007-06-06",
```

```
"fulfillment_notes": "",
"mmrrc_id": "2245",
"restriction_id": "1",
"status_dt": "2007-06-06",
"status_initials": "",
"status_notes": "BG clone, not required",
"status_user": "jdoe",
"icc_restriction_oid": "",
"icc_strain_oid": "",
"mmrrc_order_oid": "7827472",
"mta_on_file_oid": "",
"status_cd": "N/A",
"create_by": "267851",
"create_dt": "2007-06-06",
"mod_by": "270818",
"mod_dt": "2007-08-22",
"restriction_name": "Non-Profit Only<sup>1</sup>",
"is_patent": "0",
"restriction_desc": "The donor or their institution...",
"expiration_dt": "",
"nonprofit_only": "1",
"reqd_of_nonprofit": "0",
"holder": "",
"contact": "Contact Donating Institution",
"terms": "",
"is_standard": "1",
"show_publicly": "1",
"is_mta": "0",
"url": "",
"strain_name": "BayGenomics ES cell line RRF531"
}],
"items": [{
  "order_item_oid": "7827483",
  "is_active": "1",
  "item_price": "225",
  "item_status_cd": "Active",
  "needed_by_dt": "",
  "qty": "1",
  "item_master_oid": "7663313",
  "mmrrc_order_oid": "7827472",
  "order_item_rfk": "",
  "create_by": "267447",
  "create_dt": "2007-05-29",
  "mod_by": "267447",
  "mod_dt": "2010-02-21",
  "age_desc": "",
  "alt_item_code": "RRF531",
  "ctr_item_code": "",
  "gender": "",
  "icc_item_code": "002245-UCD-CELL",
  "is_available": "1",
  "item_desc": "BayGenomics ES cell line RRF531",
```

```
"product_note_oid": "",
"strain_archive_oid": "7663312",
"material_format": "ES cell line",
"fulfillment_time_tx": "Typically 2 to 4 weeks with...",
"item_mstr_fee_category_oid": "10974224 ",
"strain_name": "BayGenomics ES cellline RRF531",
"mmrrc_id": "2245",
"standard_uom": "Vial",
"close_status": "AVD",
"specs": [{
  "order_item_spec_optn_oid": "7827484",
  "desc_tx": "ES cell line genetic confirmation",
  "price": "0",
  "value_tx": "No",
  "item_spec_oid": "7340132",
  "order_item_oid": "7827483",
  "create_by": "",
  "create_dt": "",
  "mod_by": "267447",
  "mod_dt": "2011-03-30",
  "center_id": "UCD",
  "html_display_type": "select",
  "icc_center_oid": "",
  "product_note_oid": "",
  "applies_to_format": "Cell Line",
  "spec_type": "Resequencing",
  "item_spec_fee_category_oid": "10974454",
  "icc_item_code_fk": ""
}, {
  "order_item_spec_optn_oid": "7827485",
  "desc_tx": "Micro-Injection Service",
  "price": "0",
  "value_tx": "No",
  "item_spec_oid": "7340133",
  "order_item_oid": "7827483",
  "create_by": "",
  "create_dt": "",
  "mod_by": "267447",
  "mod_dt": "2010-02-21",
  "center_id": "UCD",
  "html_display_type": "select",
  "icc_center_oid": "",
  "product_note_oid": "",
  "applies_to_format": "Cell Line",
  "spec_type": "MicroInjection",
  "item_spec_fee_category_oid": "10974451",
  "icc_item_code_fk": ""
}, {
  "order_item_spec_optn_oid": "7827486",
  "desc_tx": "Parental line E14Tg2a.4 <a href=\"...",
  "price": "0",
  "value_tx": "No",
```

```
    "item_spec_oid": "7506651",
    "order_item_oid": "7827483",
    "create_by": "267447",
    "create_dt": "2007-04-10",
    "mod_by": "267447",
    "mod_dt": "2010-02-21",
    "center_id": "UCD",
    "html_display_type": "select",
    "icc_center_oid": "",
    "product_note_oid": "",
    "applies_to_format": "Cell Line",
    "spec_type": "Parental",
    "item_spec_fee_category_oid": "10974452",
    "icc_item_code_fk": ""
  }
}],
"shipments": [{
  "item_master_oid": "7663313",
  "age_desc": "",
  "alt_item_code": "RRF531",
  "ctr_item_code": "",
  "gender": "",
  "icc_item_code": "002245-UCD-CELL",
  "is_available": "1",
  "item_desc": "BayGenomics ES cell line RRF531",
  "product_note_oid": "",
  "strain_archive_oid": "7663312",
  "create_by": "267447",
  "create_dt": "2007-05-29",
  "mod_by": "267447",
  "mod_dt": "2010-02-21",
  "material_format": "ES cell line",
  "fulfillment_time_tx": "Typically 2 to 4 weeks with...",
  "item_mstr_fee_category_oid": "10974224",
  "qty": "1",
  "partial_ship_dt": "2007-03-05",
  "shipped_item_oid": "8127254",
  "note_tx": ""
}],
"milestones": "",
"notes": [{
  "order_note_oid": "7827488",
  "note_dt": "2007-06-06",
  "note_tx": "2/16/07 request date",
  "note_type_cd": "Customer Submission",
  "user_id": "",
  "mmrrc_order_oid": "7827472",
  "create_by": "267851",
  "create_dt": "2007-06-06",
  "mod_by": "",
  "mod_dt": "",
  "Name": ""
}]
```

```
    }],  
    "funding": "",  
    "nonprofit_status": "NP",  
    "customer_label": "",  
    "grant_numbers": ""  
  }  
}
```

List of PUT functions

updateOrderStatus()

This function updates order status.

Endpoint: /api/v1/orders/updateOrderStatus **Example request (PUT):** [https://api.mmrrc.org /api/v1/orders/updateOrderStatus](https://api.mmrrc.org/api/v1/orders/updateOrderStatus)

Example JSON input:

```
{  
  "order_id": "146694018",  
  "state_id": "3201"  
}
```

Example response:

```
{  
  "success": "Order Status Updated Successfully to 3200"  
}
```

List of POST functions

addOrderNote()

This function adds a user note to an order.

Endpoint: /api/v1/orders/addOrderNote

Example request (POST): <https://api.mmrrc.org/api/v1/orders/addOrderNote>

Example JSON input:

```
{
  "order_id": "146694018",
  "note": "This is note text."
}
```

Example response:

```
{
  "success": "Note Successfully added to 146694018"
}
```

addOrderMilestone()

This function adds a milestone to an order.

Endpoint: /api/v1/orders/addOrderMilestone

Example request (POST): <https://api.mmrrc.org/api/v1/orders/addOrderMilestone>

Example JSON input:

```
{
  "order_id": "146694018",
  "milestone": "Colony expansion started",
  "completed_dt": "2017-03-28",
  "note": "This is note text."
}
```

Example response:

```
{
  "success": "Milestone Successfully added to 146694018"
}
```

addOrderShipment()

This function adds an order shipment.

Endpoint: /api/v1/orders/addOrderShipment

Example request (POST): <https://api.mmrrc.org/api/v1/orders/addOrderShipment>

Example JSON input:

```
{
  "order_id": "146694018",
  "item_code": "042212-UCD-HET-M",
  "note": "Shipping might come in early.",
  "ship_dt": "2017-03-07",
  "qty": "1125"
}
```

Example response:

```
{
  "success": "Shipment successfully added to order 146694018"
}
```

Creating a new submission

Note: accessing this resource requires authentication.

Description: This resource enters a strain submission into the curation process and returns mmrrc_id.

Endpoint: /api/v1/submission/

Example request (POST): <https://api.mmrrc.org/api/v1/submission/>

Example response:

```
{
  "success":
  {
    "mmrrc_id": "51262",
    "locus_id":
    [
      "67091",
      "45044",
      "66399"
    ],
    "genetic_alteration_id": "53239"
  }
}
```

Updating an existing submission

Updates only need to send the fields that are being changed and the

`importation_data center_id` that matches the API key being used. The response will include a change entry for each field that was changed.

Endpoint: /api/v1/submission/updateSubmission/

Example request (PUT): <https://api.mmrrc.org/api/v1/submission/updateSubmission/64703>

Example JSON input:

```
{
  "mta_restrictions": [
    "KOMP MTA",
    "MMRRC-COU"
  ],
  "importation_data": {
    "center_id": "UCD"
  }
}
```

Example response:

```
{
  "success": {
    "mmrrc_id": "64703",
    "locus_id": [],
    "genetic_alteration_id": "",
    "changes": [
      {
        "mmrrc_id": "64703",
        "change_log_type_id": "2",
        "field_name": "mta_restrictions",
        "field_label": "MTA Restrictions",
        "changed_from": "MMRRC-COU",
        "changed_to": "",
        "created_by": "13716",
        "created_dt": "2018-12-21 14:24:06"
      }
    ]
  }
}
```

Caveats:

- All interactions with the API are logged, including IP addresses for anonymous connections and center credentials for authenticated usage.
- All submissions will have a status of “APIsubmission” post API validation and will not be publicly available until the data is QC’d and curated.
- We will be modifying this API based on user feedback so please feel free to contact us with requests. We will update this documentation to reflect changes as we can.

Useful Tools:

HTML Response

Retrieval (GET) functions accept 'HTML' as the last parameter in the URL. If this parameter is present, the output will be returned in the HTML format for easy reading, instead of the JSON. Example usage:

```
GET https://api.mmrrc.org/api/v1/catalog/getCatalog/20/html
```

JSON linter

To make sure that the input JSON is valid or to beautify the JSON returned by the API, you can use one of many free online JSON linters such as: jsonlint.com or codebeautify.org.

cURL

To easily test the API from the command line, use cURL. The following example creates a new submission via the API:

```
$ curl --header "x-api-key: eceab84fdb2d986f" --data-binary @sample_api.json  
http://api.mmrrc.org/api/v1/submission
```

where:

- `eceab84fdb2d986f` is your private API key,
- `sample_api.json` is the name of the file containing JSON input (see examples above).