# Facilitating access to physical material

Matija Obreza, Crop Trust, May 2020

**FIRST DRAFT**

Genebanks follow national and international regulation and internal distribution policies for incoming requests for material. They commonly provide online ordering systems where requestors not only submit a list of the accessions they want, but also provide the additional details required by the genebank to evaluate and fulfill the request.

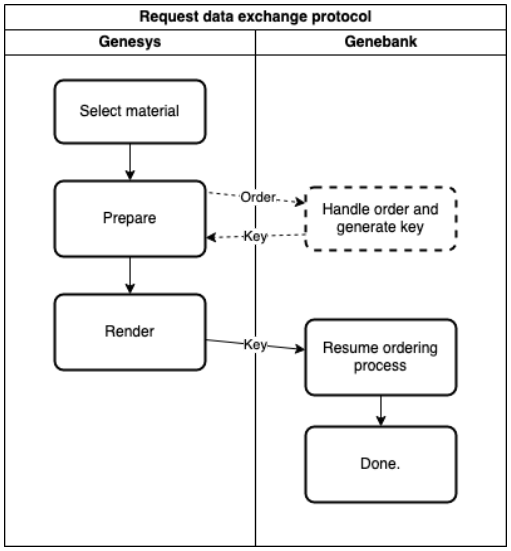
However, data about material maintained in genebanks is increasingly available through various other systems and websites not under direct control of the genebank. The availability of information in this way does not directly translate to access to physical material due to the requirements of individual genebanks.

For example, data on a genebank's collection may be accessible not only on the genebank's own website, but also through various crop portals, genomic databases like Germinate, and aggregators such as EURISCO, Genesys and the GLIS DOI portal. Potential users of germplasm (researchers and breeders) can find material of interest on any of these systems, but can generally file a request only on the genebank's website. As a result, the user is unable to conveniently issue a request for selected material when using tools on external systems; and the genebank may miss out on the opportunity to fulfil its mission of providing material to users.

This document outlines a protocol that gives genebanks the advantage of sharing collection data as widely as possible, reaching a wider group of potential users, while retaining control of the process of requesting material. The protocols described do not change the ordering system of a genebank and aim for minimal software changes in their implementation.

## A basic request data exchange protocol

The basic request data exchange protocol described here enables the external system to relay request details to the ordering systems of genebanks for further processing wherein the User can provide specific information required by the genebank.




The User has made a selection of material on the external system and initiates the request for that material.

The external system makes a HTTP POST request with order information in the request body to the specific URL on the appropriate genebank's website. The URL handler on the genebank's website reads the order data and **generates a key** associated with this order. The order **key is returned** in the HTTP response to the external system.

The external system stores the returned key with the original request record and provides the User with a

hyperlink to the target genebank's website, including the **provided key**.

The User opens the hyperlink, navigates to the genebank website, and completes the ordering process.

The basic protocol does not require extensive upgrading of the genebank's ordering system. A single new webpage (URL handler) is needed and it must handle two types of incoming HTTP requests:

1. HTTP **POST** handler receives order data, records relevant information to storage and returns a **key** with which the ordering process can be resumed.
2. HTTP **GET** handler receives the **key** as a query string parameter, loads relevant information from storage and redirects the User to the shopping cart page.

Implementation note: the POST request is made by the external system (not by the User), therefore any data needed to resume the order process cannot be stored in a HTTP session variable. Storing it in a database or a server-side memory cache will work.

## Authenticating external systems

To authenticate POST requests hitting the URL handler, the external system must include an authentication token. The authentication token must be unique for each external system. There are various options:

a) The token is provided as "Authentication: Token xxxx" HTTP Header
b) The token is provided as a form-encoded "TOKEN" field
c) The token is provided as a query string parameter

Implementation note: Use HTTPS only.

## Order data

Order data *must* include the list of selected accessions: external systems are not aware of internal identifiers in the genebank's database. For each accession in the list, the following data must be submitted: INSTCODE, ACCENUMB and DOI.

| Item | INSTCODE | ACCENUMB | DOI | |
|---|---|---|---|---|
| 1 | NLD037 | CGN00008 | 10.18730/12X68 | |
| 2 | NLD037 | CGN04491 | 10.18730/17FAP | |
| 3 | NLD000 | FOO111 | | |

The ordering system will want to check the local accession database by DOI first, then by ACCENUMB and INSTCODE. Accessions that exist in the local system are recorded and will be used when the User resumes the ordering process using the order key.

Order data *may* include *personally identifiable information*.

SMTP is more problematic than HTTPS. PII is never relayed to the target site.

## Order key

The target genebank's ordering system must ensure that the generated order key is sufficiently complex.

Implementation note: Avoid sequential keys (such as database-generated numerical identifiers) that are easy to guess. A random UUID should be sufficient.

## Order data format

The preferred format for data exchange is JSON.

Example JSON submitted by external system:

```
{
    "items": [
        { "instCode": "NLD037", "accessionNumber": "CGN00008", "doi": "10.18730/12X68" },
        { "instCode": "NLD037", "accessionNumber": "CGN04491", "doi": "10.18730/17FAP" },
        { "instCode": "NLD000", "accessionNumber": "FOO111" }
    ],
    "pii": {
        "pid": "123123",
        "name": "My name",
        "email": "email@address",
    }
}
```

## Super simple impl

HTTP **POST** /externalrequest.foo  # Receive JSON, store, return key

Response content-type: text/plain

If response code is not 2xx, fail.

If response body is blank, no items matched.

HTTP **GET** /externalrequest.foo**?key=**asda-2332-aasdsadd-... # do magic and redirect to shopping cart

Data stored in external system: The request processing URL httpx://cgn.website.nl/external.asp

Implement a simple test method.

## Resuming the ordering process

The external system will use the generated order **key** to render the hyperlink to the genebank's ordering system for the User to follow **immediately**. It may also send this hyperlink in a delayed email notification to the User to complete the ordering process.

The "resume order process hyperlink" includes the order **key** obtained from the ordering system. This is a URL for a HTTP GET request. The URL handler code of the genebank must:

1. Check that the key is valid and order data for this key exists.
    a. If order data for the key does not exist, return an error message "Order key is not valid".

2. Update the User's shopping cart
    a. Optionally clear the User's shopping cart.
    b. Populate the User's shopping cart with items listed in the order data.
3. Redirect the User to the shopping cart page or the "Checkout" page.

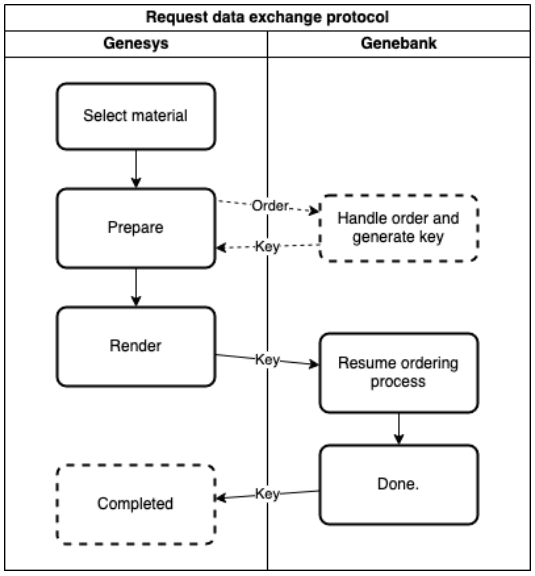
At this stage, the User can use her credentials to login to the ordering system.

Implementation note: This GET request may be triggered by antivirus/antimalware software. The handler should not immediately clear order data from storage as the hyperlink may be opened twice.

Implementation note: Order keys and associated data should be removed from storage after a timeout period of one week.

## An advanced request data exchange protocol

The advanced version of the protocol extends the basic protocol by informing Genesys of changes to the status of order processing.




The genebank's ordering system software must obtain a Client ID and Secret from Genesys and use these to make authorized API calls that affect order status changes in Genesys.

With up-to-date request status information, Genesys can "close" the request and be assured that the order has been relayed to the target genebank for processing.