

# Guide to Transfer and Restore API Functions - Plan the Transfer Session

[<< Back - Introduction](#)

[Next - Create the Transfer Session >>](#)

## Transfer and Restore API Functions

- Plan session
- Create session
- Check session
- Queue the transfer item
- Start session
- Monitor session
- Pause and restart session

## Overview

**Note:**

If you already know the modules that you need and the schema of each module, you can skip this step.

The planning process sets up the transfer session.

- For `root` sessions, retrieve the transfer modules and schema.
- For user transfer sessions, use the `AccountRemoteUser` user-based transfer session .

In these examples, the following statements are true:

- You have two servers: `remote.example.com` and `hostname.example.com`
- You logged in to the local server with the `cpsess#####` security token.
- The `example` account owns `example.com`, and you want to transfer it from the remote server to the local server.
- The `example.com` domain is 123,456,789 bytes.
- You want to transfer the `package1` package from the remote server to the local server.
- The `root` password for the remote server is `luggage12345`

## Retrieve the transfer modules

To retrieve a list of the transfer modules available on your server, run the `available_transfer_modules` function as in the following examples:

JSON API PHP

```
https://hostname.example.com:2087/cpsess#####/json-api/available_transfer_modules?api.version=1
```

```

include_once '../xmlapi.php';

$ip = getenv('REMOTE_HOST');
$root_user = getenv('REMOTE_USER');
$root_pass = getenv('REMOTE_PASSWORD');

$xmlapi = new xmlapi($ip , $root_user , $root_pass);
$xmlapi->set_output('json');

$settings = array('api.version'=>1);
$_result = $xmlapi->xmlapi_query('available_transfer_modules' , $settings);

print $_result;

```

This example will return the following list of transfer modules:

▼ [Click to view...](#)

```

{
  "data": {
    "modules": {
      "LegacyAccountBackup": "6000",
      "FeatureListRemoteRoot": "1000",
      "PackageRemoteRoot": "2000",
      "AccountLocal": "5000",
      "AccountRemoteRoot": "3000",
      "AccountRemoteUser": "4000"
    }
  },
  "metadata": {
    "version": 1,
    "reason": "OK",
    "result": "1",
    "command": "available_transfer_modules"
  }
}

```

You can preserve features and modules between transfers. Use the following features to transfer features, module lists, and accounts:

- `FeatureListRemoteRoot` — Transfers the feature lists.
- `PackageRemoteRoot` — Transfers packages.
- `AccountRemoteRoot` — Transfers the accounts.

The schema of each transfer module formats the data for transfer.

## Retrieve the schema

The `transfer_module_schema` function retrieves the following schemas:

`JSON APIPHP`

To retrieve the schema for `FeatureListRemoteRoot`, run the following browser-based call:

```
https://hostname.example.com:2087/cpsess#####/json-api/transfer_module_schema?api.version=1&module=FeatureListsRemoteRoot
```

To retrieve the schema for `PackageRemoteRoot`, run the following browser-based call:

```
https://hostname.example.com:2087/cpsess#####/json-api/transfer_module_schema?api.version=1&module=PackageRemoteRoot
```

To retrieve the schema for `AccountRemoteRoot`, run the following browser-based call:

```
https://hostname.example.com:2087/cpsess#####/json-api/transfer_module_schema?api.version=1&module=AccountRemoteRoot
```

To retrieve the schema for `FeatureListRemoteRoot`, add the following code to your PHP script :

▼ [Click to view...](#)

```
include_once '../xmlapi.php';

$ip = getenv('REMOTE_HOST');
$root_user = getenv('REMOTE_USER');
$root_pass = getenv('REMOTE_PASSWORD');

$xmlapi = new xmlapi($ip , $root_user , $root_pass);
$xmlapi->set_output('json');

$settings = array('api.version'=>1, 'module' =>
'FeatureListRemoteRoot');
$_result = $xmlapi->xmlapi_query('transfer_module_schema' , $settings);

print $_result;
```

To retrieve the schema for `PackageRemoteRoot`, add the following code to your PHP script :

▼ [Click to view...](#)

```

include_once '../xmlapi.php';

$ip = getenv('REMOTE_HOST');
$root_user = getenv('REMOTE_USER');
$root_pass = getenv('REMOTE_PASSWORD');

$xmlapi = new xmlapi($ip , $root_user , $root_pass);
$xmlapi->set_output('json');

$settings = array('api.version'=>1, 'module' => 'PackageRemoteRoot');
$_result = $xmlapi->xmlapi_query('transfer_module_schema' , $settings);

print $_result;

```

To retrieve the schema for AccountRemoteRoot, add the following code to your PHP script:

▼ [Click to view...](#)

```

include_once '../xmlapi.php';

$ip = getenv('REMOTE_HOST');
$root_user = getenv('REMOTE_USER');
$root_pass = getenv('REMOTE_PASSWORD');

$xmlapi = new xmlapi($ip , $root_user , $root_pass);
$xmlapi->set_output('json');

$settings = array('api.version'=>1, 'module' => 'AccountRemoteRoot');
$_result = $xmlapi->xmlapi_query('transfer_module_schema' , $settings);

print $_result;

```

The schema will resemble the following example:

▼ [Click to view...](#)

```

{
  "data": {
    "schema": {
      "keys": {
        "detected_remote_user": {
          "def": "char(255) DEFAULT NULL"
        },
        "ip": {
          "def": "int(1) DEFAULT 0"
        },
        "skipbwdata": {
          "def": "int(1) DEFAULT 0"
        },
        "customip": {
          "def": "char(255) DEFAULT NULL"
        },
      },
    },
  },
}

```

```
"size": {
  "def": "BIGINT UNSIGNED DEFAULT 1"
},
"shared_mysql_server": {
  "def": "int(1) DEFAULT 0"
},
"domain": {
  "def": "char(255) DEFAULT NULL"
},
"cpmovefile": {
  "def": "text"
},
"skiphomedir": {
  "def": "int(1) DEFAULT 0"
},
"user": {
  "def": "char(255) DEFAULT NULL"
},
"skipacctdb": {
  "def": "int(1) DEFAULT 0"
},
"skipres": {
  "def": "int(1) DEFAULT 0"
},
"reseller": {
  "def": "int(1) DEFAULT 0"
},
"skipaccount": {
  "def": "int(1) DEFAULT 0"
},
"copypoint": {
  "def": "text"
},
"force": {
  "def": "int(1) DEFAULT 0"
},
"xferpoint": {
  "def": "int(1) DEFAULT 0"
},
"replaceip": {
  "def": "char(255) DEFAULT NULL"
},
"localuser": {
  "def": "char(255) DEFAULT NULL"
}
},
"required": [
  "user",
  "localuser"
],
"primary": [
  "user"
]
]
```

```
}  
},  
"metadata": {  
  "version": 1,  
  "reason": "OK",  
  "result": "1",
```

```
    "command": "transfer_module_schema"  
  }  
}
```

## Additional documentation

[Suggested documentation](#) [For cPanel users](#) [For WHM users](#) [For developers](#)

- [Guide to Transfer and Restore API Functions - Plan the Transfer Session](#)
- [Guide to Transfer and Restore API Functions - Create the Transfer Session](#)
- [Guide to Transfer and Restore API Functions - Check the Transfer Session](#)
- [Guide to Transfer and Restore API Functions - Queue the Transfer Item](#)
- [Guide to Transfer and Restore API Functions - Start the Transfer Session](#)

Error rendering macro 'contentbylabel' : parameters should not be empty

### Content by label

There is no content with the specified labels



### Content by label

There is no content with the specified labels

