

BellSoft Product Discovery API



Liberica JDK
Revision 1.1
February 2025

be//soft

Copyright © BellSoft Corporation 2018-2025.

BellSoft software contains open source software. Additional information about third party code is available at https://bell-sw.com/third_party_licenses. You can also get more information on how to get a copy of source code by contacting info@bell-sw.com.

THIS INFORMATION MAY CHANGE WITHOUT NOTICE. TO THE EXTENT PERMITTED BY APPLICABLE LAW, BELLSOFT PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL BELLSOFT BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF BELLSOFT IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in this document is governed by the applicable license agreement, which is not modified in any way by the terms of this notice.

Alpaquita, Liberica and BellSoft are trademarks or registered trademarks of BellSoft Corporation. The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Java and OpenJDK are trademarks or registered trademarks of Oracle and/or its affiliates. Other trademarks are the property of their respective owners and are used only for identification purposes.

Contents

1. Overview	6
2. Operating systems, architectures and packages	7
Architectures	7
Operating systems	7
3. Discover the latest Liberica JDK release	9
4. Discover the latest Liberica JDK LTS release	11
5. Check if your Liberica JDK version is the latest version	13
By Filename	13
By Version	14
6. Get a download link without parsing JSON	16

7. Discover the list of the Liberica NIK components	17
8. Discover the list of the Liberica NIK bundle types	18
9. Discover the general Liberica NIK component set	19
10. Discover the general Liberica NIK component set with selected fields	22
11. Discover src.tar.gz Liberica NIK component set	28
12. Discover the base Liberica NIK component set	29
13. Discover the base and nested Liberica NIK component sets together	31
14. Discover the base Liberica NIK component set	

that contains the nested component with the

required version only

34

15. Discover the nested Liberia NIK component set

only

38

16. Discover the base and nested Liberia NIK

component sets together - getting a text mode

response

41

17. Discover the nested Liberia NIK component set

only - getting a text mode response

45

1. Overview

BellSoft Product Discovery REST API helps users discover and query for BellSoft products (specifically, Liberica JDK) download URLs, versions, architectures, and features programmatically.

This document lists examples to solve common tasks with a REST Product Discovery API. For more technical details, please refer to the Product Discovery OpenApi (Swagger) specification. See [OpenAPI Specification](#) for more details.

2. Operating systems, architectures and packages

The BellSoft product discovery API has several methods to discover lists of supported operating systems and architectures.

Architectures

Url: <https://api.bell-sw.com/v1/liberica/architectures>

JSON response example:

```
["arm", "ppc", "sparc", "x86"]
```

Bash example:

```
curl https://api.bell-sw.com/v1/liberica/architectures
```

PowerShell example:

```
Invoke-RestMethod https://api.bell-sw.com/v1/liberica/architectures
```

Operating systems

Url: <https://api.bell-sw.com/v1/liberica/operating-systems>

JSON response example:

```
["linux", "linux-musl", "macos", "solaris", "windows"]
```

Bash example:

```
curl https://api.bell-sw.com/v1/liberica/operating-systems
```

PowerShell example:

```
Invoke-RestMethod https://api.bell-sw.com/v1/liberica/operating-systems
```


3. Discover the latest Liberica JDK release

In this example, we assume that the user knows all user-specific options, such as bitness, os, etc. So the list of releases should be very short.

Url: <https://api.bell-sw.com/v1/liberica/releases?version-modifier=latest&bitness=64&os=windows&arch=x86&package-type=zip&bundle-type=jdk>

JSON response example:

```
[
  {
    "bitness": 64,
    "buildVersion": 10,
    "latestLTS": false,
    "os": "windows",
    "updateVersion": 1,
    "downloadUrl": "https://github.com/bell-
sw/Liberica/releases/download/13.0.1+10/BellSoft-jdk13.0.1+10-windows-
amd64.zip",
    "interimVersion": 0,
    "EOL": false,
    "latestInFeatureVersion": true,
    "LTS": false,
    "bundleType": "jdk",
    "version": "13.0.1+10",
    "featureVersion": 13,
    "packageType": "zip",
    "sha1": "a2143878d68f67e4cd079d52365b415451c4bf15",
    "FX": true,
    "filename": "BellSoft-jdk13.0.1+10-windows-amd64.zip",
    "installationType": "archive",
    "size": 275301289,
    "patchVersion": 0,
    "GA": true,
    "architecture": "x86",
    "latest": true
  }
]
```

```
}  
]
```

Bash example:

```
curl https://api.bell-sw.com/v1/liberica/releases?version-  
modifier=latest&bitness=64&os=windows&arch=x86&package-type=zip&bundle-type=jdk
```

PowerShell example:

```
Invoke-RestMethod https://api.bell-sw.com/v1/liberica/releases?version-  
modifier=latest&bitness=64&os=windows&arch=x86&package-type=zip&bundle-type=jdk
```

4. Discover the latest Liberica JDK LTS release

As in the previous example, we will limit the output using additional user-specific filters.

Url: <https://api.bell-sw.com/v1/liberica/releases?version-modifier=latest&bitness=64&release-type=lts&os=windows&arch=x86&package-type=zip&bundle-type=jdk>

JSON response example:

```
[
  {
    "bitness": 64,
    "buildVersion": 11,
    "latestLTS": true,
    "os": "windows",
    "updateVersion": 5,
    "downloadUrl": "https://github.com/bell-
sw/Liberica/releases/download/11.0.5+11/BellSoft-jdk11.0.5+11-windows-
amd64.zip",
    "interimVersion": 0,
    "EOL": false,
    "latestInFeatureVersion": true,
    "LTS": true,
    "bundleType": "jdk",
    "version": "11.0.5+11",
    "featureVersion": 11,
    "packageType": "zip",
    "sha1": "9956658bdc98f844bacbf451b8e8f0622544b539",
    "FX": true,
    "filename": "BellSoft-jdk11.0.5+11-windows-amd64.zip",
    "installationType": "archive",
    "size": 265347010,
    "patchVersion": 0,
    "GA": true,
    "architecture": "x86",
    "latest": false
  }
]
```

]

Bash example:

```
curl https://api.bell-sw.com/v1/liberica/releases?version-  
modifier=latest&bitness=64&release-type=lts&os=windows&arch=x86&package-  
type=zip&bundle-type=jdk
```

PowerShell example:

```
Invoke-RestMethod https://api.bell-sw.com/v1/liberica/releases?version-  
modifier=latest&bitness=64&release-type=lts&os=windows&arch=x86&package-  
type=zip&bundle-type=jdk
```

5. Check if your Liberica JDK version is the latest version

By Filename

In this example, we assume that the user knows the name of the file that was previously downloaded. It is possible to check if the Liberica JDK release is the latest in feature release, or latest LTS release, or simply the latest.

Url: <https://api.bell-sw.com/v1/liberica/releases/BellSoft-jdk11.0.5+11-windows-amd64.zip>

JSON response example:

```
{
  "bitness": 64,
  "buildVersion": 11,
  "latestLTS": true,
  "os": "windows",
  "updateVersion": 5,
  "downloadUrl": "https://github.com/bell-
sw/Liberica/releases/download/11.0.5+11/BellSoft-jdk11.0.5+11-windows-
amd64.zip",
  "interimVersion": 0,
  "EOL": false,
  "latestInFeatureVersion": true,
  "LTS": true,
  "bundleType": "jdk",
  "version": "11.0.5+11",
  "featureVersion": 11,
  "packageType": "zip",
  "sha1": "9956658bdc98f844bacbf451b8e8f0622544b539",
  "FX": true,
  "filename": "BellSoft-jdk11.0.5+11-windows-amd64.zip",
  "installationType": "archive",
  "size": 265347010,
```

```
"patchVersion": 0,  
"GA": true,  
"architecture": "x86",  
"latest": false  
}
```

Bash example:

```
curl https://api.bell-sw.com/v1/liberica/releases/BellSoft-jdk11.0.5+11-  
windows-amd64.zip
```

PowerShell example:

```
Invoke-RestMethod https://api.bell-sw.com/v1/liberica/releases/BellSoft-  
jdk11.0.5+11-windows-amd64.zip
```

By Version

In this example, we assume that the user knows the version string of Liberica JDK release that was previously downloaded. It is possible to check the flags similar to the previous example.

Url: <https://api.bell-sw.com/v1/liberica/releases?version=11.0.5+11&bitness=64&os=windows&arch=x86&package-type=zip&bundle-type=jdk>

JSON response example:

```
[  
  {  
    "bitness": 64,  
    "buildVersion": 11,  
    "latestLTS": true,  
    "os": "windows",  
    "updateVersion": 5,  
    "downloadUrl": "https://github.com/bell-  
sw/Liberica/releases/download/11.0.5+11/BellSoft-jdk11.0.5+11-windows-  
amd64.zip",  
    "interimVersion": 0,  
    "EOL": false,  
    "latestInFeatureVersion": true,  
    "LTS": true,  
    "bundleType": "jdk",  
    "version": "11.0.5+11",  
  }  
]
```

```
"featureVersion": 11,  
"packageType": "zip",  
"sha1": "9956658bdc98f844bacbf451b8e8f0622544b539",  
"FX": true,  
"filename": "BellSoft-jdk11.0.5+11-windows-amd64.zip",  
"installationType": "archive",  
"size": 265347010,  
"patchVersion": 0,  
"GA": true,  
"architecture": "x86",  
"latest": false  
}  
]
```

Bash example:

```
curl https://api.bell-  
sw.com/v1/liberica/releases?version=11.0.5+11&bitness=64&os=windows&arch=x86&pa  
ckage-type=zip&bundle-type=jdk
```

PowerShell example:

```
Invoke-RestMethod https://api.bell-  
sw.com/v1/liberica/releases?version=11.0.5+11&bitness=64&os=windows&arch=x86&pa  
ckage-type=zip&bundle-type=jdk
```

6. Get a download link without parsing JSON

This example shows how to discover a link for downloading without parsing JSON.

Url: <https://api.bell-sw.com/v1/liberica/releases?version=11.0.5+11&bitness=64&os=windows&arch=x86&package-type=zip&bundle-type=jdk&output=text&fields=downloadUrl>

Response example:

```
https://github.com/bell-sw/Liberica/releases/download/11.0.5+11/{COMPANY_NAME}-jdk11.0.5+11-windows-amd64.zip
```

Bash example:

```
curl https://api.bell-sw.com/v1/liberica/releases?version=11.0.5%2B11&bitness=64&os=windows&arch=x86&package-type=zip&bundle-type=jdk&output=text&fields=downloadUrl
```

PowerShell example:

```
Invoke-RestMethod https://api.bell-sw.com/v1/liberica/releases?version=11.0.5%2B11&bitness=64&os=windows&arch=x86&package-type=zip&bundle-type=jdk&output=text&fields=downloadUrl
```


7. Discover the list of the Liberica NIK components

Liberica Native Image Kit contains several sets of **base** components, where each includes different nested components (embedded or installable). These sets differ by versions, operating systems, and processor architectures.

The **base** component is a nik component.

Any other components are nested components.

Some of them — **liberica**, **nodejs** — are embedded components.

The rest — **llvm**, **ni**, **python**, **r**, **ruby**, **wasm** — are installable (that is not embedded) components.

URL: <https://api.bell-sw.com/v1/nik/components>

JSON response example (alphabetical order):

```
["liberica","llvm","ni","nik","nodejs","python","r","ruby","wasm"]
```

Bash example:

```
curl "https://api.bell-sw.com/v1/nik/components"
```

PowerShell example:

```
Invoke-RestMethod "https://api.bell-sw.com/v1/nik/components"
```

8. Discover the list of the Liberica NIK bundle types

Liberica Native Image Kit contains several bundle types.

URL: <https://api.bell-sw.com/v1/nik/bundle-types>

JSON response example (alphabetical order):

```
["core","standard"]
```

Bash example:

```
curl "https://api.bell-sw.com/v1/nik/bundle-types"
```

PowerShell example:

```
Invoke-RestMethod "https://api.bell-sw.com/v1/nik/bundle-types"
```

9. Discover the general Liberica NIK component set

Since the result JSON response, presented as the JSON array, is massive for the general URL, the only item returned is the first one (the first base NIK component with all its nested NIK components).

URL: <https://api.bell-sw.com/v1/nik/releases>

JSON response example:

```
[
  {
    "bitness": 64,
    "extraVersion": 2,
    "EOL": false,
    "latestLTS": true,
    "os": "macos",
    "latestInAnnualVersion": true,
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-openjdk11-21.0.0.2-macos-amd64.zip",
    "LTS": true,
    "packageType": "zip",
    "featureVersion": 0,
    "version": "21.0.0.2",
    "releaseCatalogUrl": "https://download.bell-sw.com/vm/release-catalog-java11.properties",
    "sha1": "49948207b6d414ec09cfa069ccfb920cac4af436",
    "component": "nik",
    "filename": "{COMPANY_NAME}-liberica-vm-openjdk11-21.0.0.2-macos-amd64.zip",
    "annualVersion": 21,
    "installationType": "archive",
    "size": 459490970,
    "patchVersion": 0,
    "GA": true,
    "architecture": "x86",
    "latest": true,
    "components": [
```

```

{
  "embedded": true,
  "component": "liberica"
},
{
  "embedded": true,
  "component": "nodejs"
},
{
  "sha1": "3f65fd40fea70b32b240d7f319151aa87dd6af07",
  "component": "llvm",
  "filename": "llvm-toolchain-installable-openjdk11-21.0.0.2-macos-
amd64.jar",
  "size": 113000219,
  "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/llvm-
toolchain-installable-openjdk11-21.0.0.2-macos-amd64.jar",
  "packageType": "jar",
  "embedded": false
},
{
  "sha1": "52419f5ef3739f3fc3902acda4551f3645fc223f",
  "component": "python",
  "filename": "python-installable-openjdk11-21.0.0.2-macos-amd64.jar",
  "size": 429456984,
  "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-macos-amd64.jar",
  "packageType": "jar",
  "embedded": false
},
{
  "sha1": "72ac9dc0ef5106179bbbc2a57ecb90527370217c",
  "component": "ni",
  "filename": "native-image-installable-openjdk11-21.0.0.2-macos-
amd64.jar",
  "size": 40061425,
  "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/native-image-
installable-openjdk11-21.0.0.2-macos-amd64.jar",
  "packageType": "jar",
  "embedded": false
},
{
  "sha1": "59942ef3030187f75cb928ab89fd8a984cc0baf4",
  "component": "ruby",
  "filename": "ruby-installable-openjdk11-21.0.0.2-macos-amd64.jar",

```

```

    "size": 51743490,
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-macos-amd64.jar",
    "packageType": "jar",
    "embedded": false
  },
  {
    "sha1": "69172d29b1c84436ca5ffc2806e85d8795b2de1a",
    "component": "r",
    "filename": "r-installable-openjdk11-21.0.0.2-macos-amd64.jar",
    "size": 355164665,
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/r-installable-
openjdk11-21.0.0.2-macos-amd64.jar",
    "packageType": "jar",
    "embedded": false
  },
  {
    "sha1": "ab43f0e1b6dc67e364fe03cb53db6ac4ec27cdaf",
    "component": "wasm",
    "filename": "wasm-installable-openjdk11-21.0.0.2-macos-amd64.jar",
    "size": 33091667,
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/wasm-
installable-openjdk11-21.0.0.2-macos-amd64.jar",
    "packageType": "jar",
    "embedded": false
  }
]
},
...
]

```

Bash example:

```
curl "https://api.bell-sw.com/v1/nik/releases"
```

PowerShell example:

```
Invoke-RestMethod "https://api.bell-sw.com/v1/nik/releases"
```

10. Discover the general Liberica NIK component set with selected fields

In this example, the `fields=component,downloadUrl,components` request parameter is used so that the JSON response includes just the set of the mentioned fields. The `fields` request parameter does not filter the components included in the result JSON response but only defines which properties of the components will appear in the JSON response.

! Important:

The `fields` request parameter has to include the `components` value if you want the nested NIK components to be included in the result JSON response. Otherwise, the JSON response will contain only the base NIK components.

The nested NIK component `liberica` is embedded and thus has no `downloadUrl` field.

The `arch=x86` request filter (request parameter) is applied to the URL to limit the JSON response length.

URL: <https://api.bell-sw.com/v1/nik/releases?fields=component,downloadUrl,components&arch=x86>

JSON response example:

```
[
  {
    "components": [
      {
        "component": "liberica"
      },
      {
        "component": "ni",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/native-image-installable-openjdk11-21.0.0.2-linux-x64-musl.jar"
      }
    ]
  }
]
```

```

    },
    {
      "component": "ruby",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-linux-x64-musl.jar"
    },
    {
      "component": "r",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/r-installable-
openjdk11-21.0.0.2-linux-x64-musl.jar"
    },
    {
      "component": "wasm",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/wasm-
installable-openjdk11-21.0.0.2-linux-x64-musl.jar"
    },
    {
      "component": "llvm",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/llvm-
toolchain-installable-openjdk11-21.0.0.2-linux-x64-musl.jar"
    },
    {
      "component": "python",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-linux-x64-musl.jar"
    }
  ],
  "component": "nik",
  "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-
vm-openjdk11-21.0.0.2-linux-x64-musl.tar.gz"
},
{
  "components": [
    {
      "component": "liberica"
    },
    {
      "component": "ruby",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-linux-amd64.jar"
    },
    {
      "component": "r",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/r-installable-

```

```

openjdk11-21.0.0.2-linux-amd64.jar"
    },
    {
        "component": "wasm",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/wasm-
installable-openjdk11-21.0.0.2-linux-amd64.jar"
    },
    {
        "component": "llvm",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/llvm-
toolchain-installable-openjdk11-21.0.0.2-linux-amd64.jar"
    },
    {
        "component": "python",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-linux-amd64.jar"
    },
    {
        "component": "ni",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/native-image-
installable-openjdk11-21.0.0.2-linux-amd64.jar"
    }
],
"component": "nik",
"downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-
vm-openjdk11-21.0.0.2-linux-amd64.tar.gz"
},
{
    "components": [
        {
            "component": "liberica"
        },
        {
            "component": "llvm",
            "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/llvm-
toolchain-installable-openjdk11-21.0.0.2-macos-amd64.jar"
        },
        {
            "component": "python",
            "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-macos-amd64.jar"
        },
        {
            "component": "ni",

```



```

      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/native-image-
installable-openjdk11-21.0.0.2-macos-amd64.jar"
    },
    {
      "component": "ruby",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-macos-amd64.jar"
    },
    {
      "component": "r",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/r-installable-
openjdk11-21.0.0.2-macos-amd64.jar"
    },
    {
      "component": "wasm",
      "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/wasm-
installable-openjdk11-21.0.0.2-macos-amd64.jar"
    }
  ],
  "component": "nik",
  "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-
vm-openjdk11-21.0.0.2-macos-amd64.zip"
}
]

```

Bash example:

```
curl "https://api.bell-
sw.com/v1/nik/releases?fields=component,downloadUrl,components&arch=x86"
```

PowerShell example:

```
Invoke-RestMethod "https://api.bell-
sw.com/v1/nik/releases?fields=component,downloadUrl,components&arch=x86"
```

If the `components` value is not included into the `fields` request parameter, the JSON response will return no installable or embedded NIK components because the `components` property of each base NIK component is ignored.

❗ Important:

The JSON response is not filtered by any request parameters and thus includes the base NIK components and a special additional one: `src.tar.gz`. This `src.tar.gz` NIK component is neither base nor installable/embedded. In this sense, we can say it

is an unknown NIK component since it has no component property.

URL: <https://api.bell-sw.com/v1/nik/releases?fields=component,downloadUrl>

JSON response example:

```
[
  {
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-
liberica-vm-openjdk11-21.0.0.2-linux-aarch64.tar.gz"
  },
  {
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-
liberica-vm-openjdk11-21.0.0.2-linux-x64-musl.tar.gz"
  },
  {
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-
liberica-vm-openjdk11-21.0.0.2-linux-amd64.tar.gz"
  },
  {
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-
liberica-vm-openjdk11-21.0.0.2-linux-aarch64-musl.tar.gz"
  },
  {
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-
liberica-vm-openjdk11-21.0.0.2-macos-amd64.zip"
  },
  {
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-
liberica-vm-openjdk11-21.0.0.2-src.tar.gz"
  }
]
```

Bash example:

```
curl "https://api.bell-sw.com/v1/nik/releases?fields=component,downloadUrl"
```

PowerShell example:

```
Invoke-RestMethod "https://api.bell-  
sw.com/v1/nik/releases?fields=component,downloadUrl"
```

11. Discover src.tar.gz Liberica NIK component set

It is recommended to use the `package-type=src.tar.gz` request parameter to receive the required set of the `src.tar.gz` NIK components.

URL:

<https://api.bell-sw.com/v1/nik/releases?package-type=src.tar.gz&version=21.0.0.2> <https://api.bell-sw.com/v1/nik/releases?package-type=src.tar.gz&version-modifier=latest>

JSON response example:

```
[
  {
    "sha1": "e0445658667c59d214d7f8893715bd9c8d3ce1e7",
    "filename": "BellSoft-liberica-vm-openjdk11-21.0.0.2-src.tar.gz",
    "size": 275662456,
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-vm-openjdk11-21.0.0.2-src.tar.gz",
    "packageType": "src.tar.gz"
  }
]
```

Bash example:

```
curl "https://api.bell-sw.com/v1/nik/releases?package-type=src.tar.gz&version=21.0.0.2"
curl "https://api.bell-sw.com/v1/nik/releases?package-type=src.tar.gz&version-modifier=latest"
```

PowerShell example:

```
Invoke-RestMethod "https://api.bell-sw.com/v1/nik/releases?package-type=src.tar.gz&version=21.0.0.2"
Invoke-RestMethod "https://api.bell-sw.com/v1/nik/releases?package-type=src.tar.gz&version-modifier=latest"
```

12. Discover the base Liberica NIK component set

In this example, we use the `components=nik` request parameter to have only the base NIK components in the JSON response.

❗ Important:

The `fields` request parameter includes the `components` value (see **URL** below). The `components` property arrays are also shown. However, all these arrays of the nested NIK components are empty because the used `components=nik` request filter rejects any nested NIK component. Thus, no nested NIK components are included in the JSON response.

If the `fields` request parameter does not include the `components` value, the JSON response does not show any appropriate `components` property array at all.

URL: <https://api.bell-sw.com/v1/nik/releases?components=nik&fields=component,downloadUrl,components>

JSON response example:

```
[
  {
    "components": [],
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-vm-openjdk11-21.0.0.2-linux-aarch64.tar.gz"
  },
  {
    "components": [],
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-vm-openjdk11-21.0.0.2-linux-x64-musl.tar.gz"
  },
  {
```

```
    "components": [],
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-vm-openjdk11-21.0.0.2-linux-amd64.tar.gz"
  },
  {
    "components": [],
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-vm-openjdk11-21.0.0.2-linux-aarch64-musl.tar.gz"
  },
  {
    "components": [],
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-vm-openjdk11-21.0.0.2-macos-amd64.zip"
  }
]
```

Bash example:

```
curl "https://api.bell-sw.com/v1/nik/releases?components=nik&fields=component,downloadUrl,components"
```

PowerShell example:

```
Invoke-RestMethod "https://api.bell-sw.com/v1/nik/releases?components=nik&fields=component,downloadUrl,components"
```

13. Discover the base and nested Liberica NIK component sets together

In this example, we use the `components=nik,ruby,python` request parameter to have the base NIK component together with some of its nested NIK components in the JSON response. If any base NIK component does not contain the mentioned nested NIK components, its `components` property shows in the JSON response as an empty array.

⚠ Important:

If the `components` request parameter includes any nested NIK component, the `fields` request parameter will include the `components` value *implicitly* as its last value.

Please compare the two **URLs** below: they both return the same result.

URL: <https://api.bell-sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl> <https://api.bell-sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl,components>

JSON response example:

```
[
  {
    "components": [],
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-vm-openjdk11-21.0.0.2-linux-aarch64.tar.gz"
  },
  {
    "components": [
      {
        "component": "ruby",
```

```

        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-linux-x64-musl.jar"
    },
    {
        "component": "python",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-linux-x64-musl.jar"
    }
],
"component": "nik",
"downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-
vm-openjdk11-21.0.0.2-linux-x64-musl.tar.gz"
},
{
    "components": [
        {
            "component": "ruby",
            "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-linux-amd64.jar"
        },
        {
            "component": "python",
            "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-linux-amd64.jar"
        }
    ],
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-
vm-openjdk11-21.0.0.2-linux-amd64.tar.gz"
},
{
    "components": [],
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-
vm-openjdk11-21.0.0.2-linux-aarch64-musl.tar.gz"
},
{
    "components": [
        {
            "component": "python",
            "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-macos-amd64.jar"
        },
        {

```



```
        "component": "ruby",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-macos-amd64.jar"
    },
    "component": "nik",
    "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/BellSoft-liberica-
vm-openjdk11-21.0.0.2-macos-amd64.zip"
}
```

Bash example:

```
curl "https://api.bell-
sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl"
curl "https://api.bell-
sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl,
components"
```

Powershell example:

```
Invoke-RestMethod "https://api.bell-
sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl"
Invoke-RestMethod "https://api.bell-
sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl,
components"
```

14. Discover the base Liberica NIK component set that contains the nested component with the required version only

In this example, we use the following request parameters:

```
component-version=liberica@11.0.10+9
```

and

```
component-version=liberica@11.0.11+9
```

to have only the base NIK components that contain the nested Liberica NIK component with the required version in the JSON response.

The required nested NIK component and its version have to be defined as the value of the `component-version` request parameter in the following format (consisting of two parts with `@` as a delimiter): `nested-component@nested-component-version`.

The base NIK component is included in the resulting JSON response if the following conditions are met:

- The output contains the mentioned nested NIK component (the exact comparison is used)
- The value of the `version` property of this nested NIK component starts with the `nested-component-version` substring defined as the second part of the value in the `component-version` request parameter

It means that all the following expressions are correct and can be used if necessary:

```
component-version=liberica\@11
```

```
component-version=liberica\@11.0
```

```
component-version=liberica\@11.0.11
```

```
component-version=liberica\@11.0.11+9
```

**Note:**

The URLs below contain several additional request parameters (components, fields, bundle-type, os) just to minimize the JSON response length. The nested Liberica NIK component is used in this example, and all of the above applies to the other nested NIK components: nodejs, python, etc.

URL: <https://api.bell-sw.com/v1/nik/releases?component-version=liberica@11.0.10+9&components=nik,liberica&fields=component,filename,version,components&bundle-type=standard&os=macos>

JSON response example:

```
[
  {
    "components": [
      {
        "component": "liberica",
        "version": "11.0.10+9"
      }
    ],
    "component": "nik",
    "filename": "BellSoft-liberica-vm-openjdk11-21.0.0.2-macos-amd64.zip",
    "version": "21.0.0.2"
  }
]
```

Bash example:

```
curl "https://api.bell-sw.com/v1/nik/releases?component-version=liberica@11.0.10+9&components=nik,liberica&fields=component,filename,version,components&bundle-type=standard&os=macos"
```

PowerShell example:

```
Invoke-RestMethod "https://api.bell-sw.com/v1/nik/releases?component-version=liberica@11.0.10+9&components=nik,liberica&fields=component,filename,version,components&bundle-type=standard&os=macos"
```

URL: <https://api.bell-sw.com/v1/nik/releases?component-version=liberica@11.0.11+9&components=nik,liberica&fields=component,filename,version,components&bundle-type=standard&os=macos>

```
type=standard&os=macos
```

JSON response example:

```
[
  {
    "components": [
      {
        "component": "liberica",
        "version": "11.0.11+9"
      }
    ],
    "component": "nik",
    "filename": "BellSoft-liberica-vm-openjdk11-21.1.0-macos-amd64.dmg",
    "version": "21.1.0"
  },
  {
    "components": [
      {
        "component": "liberica",
        "version": "11.0.11+9"
      }
    ],
    "component": "nik",
    "filename": "BellSoft-liberica-vm-openjdk11-21.1.0-macos-amd64.pkg",
    "version": "21.1.0"
  },
  {
    "components": [
      {
        "component": "liberica",
        "version": "11.0.11+9"
      }
    ],
    "component": "nik",
    "filename": "BellSoft-liberica-vm-openjdk11-21.1.0-macos-amd64.zip",
    "version": "21.1.0"
  }
]
```

Bash example:

```
curl "https://api.bell-sw.com/v1/nik/releases?component-  
version=liberica@11.0.11+9&components=nik,liberica&fields=component,filename,ve
```

rsion,components&bundle-type=standard&os=macos"

Powershell example:

```
Invoke-RestMethod "https://api.bell-sw.com/v1/nik/releases?component-  
version=liberica@11.0.11+9&components=nik,liberica&fields=component,filename,ve  
rsion,components&bundle-type=standard&os=macos"
```

15. Discover the nested Liberica NIK component set only

In this example, we use the `components=ruby,python` request parameter to have only the nested NIK components in the JSON response. In this case, no property of the parent base NIK components will be included in the JSON response, except for the `components` property array. And if certain base NIK components do not contain the aforementioned *nested* NIK components, these base NIK components will certainly be excluded from the JSON response.

! Important:

If the `components` request parameter includes any nested NIK component, the `fields` request parameter will include the `components` value *implicitly* as its last value.

Please compare the two **URLs** below: they both return the same result.

URL: <https://api.bell-sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl> <https://api.bell-sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl,components>

JSON response example:

```
[
  {
    "components": [
      {
        "component": "ruby",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-linux-x64-musl.jar"
      },
      {
        "component": "python",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-linux-x64-musl.jar"
      }
    ]
  }
]
```

```

    ]
  },
  {
    "components": [
      {
        "component": "ruby",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-linux-amd64.jar"
      },
      {
        "component": "python",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-linux-amd64.jar"
      }
    ]
  },
  {
    "components": [
      {
        "component": "python",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/python-
installable-openjdk11-21.0.0.2-macos-amd64.jar"
      },
      {
        "component": "ruby",
        "downloadUrl": "https://download.bell-sw.com/vm/21.0.0.2/ruby-
installable-openjdk11-21.0.0.2-macos-amd64.jar"
      }
    ]
  }
]

```

Bash example:

```

curl "https://api.bell-
sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl"
curl "https://api.bell-
sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl,comp
onents"

```

PowerShell example:

```

Invoke-RestMethod "https://api.bell-
sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl"
Invoke-RestMethod "https://api.bell-

```

`sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl,components"`

16. Discover the base and nested LibERICA NIK component sets together - getting a text mode response

In this example, we use the `components=nik,ruby,python` request parameter to have the base NIK component together with some of its nested NIK components in the text response.

The JSON response is converted into the text response by applying the following rules to this JSON:

- All items in the text response are separated by a comma
- The order of items in the text response corresponds to the order of values in the `fields` request parameter
- For each NIK component
 - No NIK component property names are printed, only NIK component property values
 - The only NIK component property values printed are those which property names are mentioned in the `fields` request parameter
 - If the NIK component does not have the appropriate property, the value of this property is replaced by an empty string
- If the `components` property of the base NIK component is an empty array (this base NIK component has no nested NIK components), this empty array is replaced by the sequence of empty strings equal to the number of property names in the `fields` request parameter (except for the `components` property name itself)
- If the base NIK component has no nested NIK components, it is displayed in the text response as a single line
- If the base NIK component contains the nested NIK components (the `components` property of this base NIK component is a non-empty array), it is presented in the text response as several lines, equal to the number of the nested NIK components

! Important:

If the components request parameter includes any nested NIK component, the fields request parameter includes the components value implicitly as its last value.

Please compare the two URLs below: they both return the same result.

URL: <https://api.bell-sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl&output=text> <https://api.bell-sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl,components&output=text>

Text response example:

```

nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-linux-aarch64.tar.gz,,
nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-linux-x64-musl.tar.gz,ruby,https://download.bell-
sw.com/vm/21.0.0.2/ruby-installable-openjdk11-21.0.0.2-linux-x64-musl.jar
nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-linux-x64-musl.tar.gz,python,https://download.bell-
sw.com/vm/21.0.0.2/python-installable-openjdk11-21.0.0.2-linux-x64-musl.jar
nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-linux-amd64.tar.gz,ruby,https://download.bell-
sw.com/vm/21.0.0.2/ruby-installable-openjdk11-21.0.0.2-linux-amd64.jar
nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-linux-amd64.tar.gz,python,https://download.bell-
sw.com/vm/21.0.0.2/python-installable-openjdk11-21.0.0.2-linux-amd64.jar
nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-linux-aarch64-musl.tar.gz,,
nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-macos-amd64.zip,python,https://download.bell-
sw.com/vm/21.0.0.2/python-installable-openjdk11-21.0.0.2-macos-amd64.jar
nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-macos-amd64.zip,ruby,https://download.bell-
sw.com/vm/21.0.0.2/ruby-installable-openjdk11-21.0.0.2-macos-amd64.jar

```

Bash example:

```

curl "https://api.bell-
sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl&
output=text"
curl "https://api.bell-

```

```
sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl,
components&output=text"
```

PowerShell example:

```
Invoke-RestMethod "https://api.bell-
sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl&
output=text"
Invoke-RestMethod "https://api.bell-
sw.com/v1/nik/releases?components=nik,ruby,python&fields=component,downloadUrl,
components&output=text"
```

See how the text response differs with the following changes made to the `fields` request parameter: the `components` value is explicitly defined as the first value.

URL: <https://api.bell-sw.com/v1/nik/releases?components=nik,ruby,python&fields=components,component,downloadUrl&output=text>

Text response example:

```
,,nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-linux-aarch64.tar.gz
ruby,https://download.bell-sw.com/vm/21.0.0.2/ruby-installable-openjdk11-
21.0.0.2-linux-x64-musl.jar,nik,https://download.bell-
sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-openjdk11-21.0.0.2-linux-x64-
musl.tar.gz
python,https://download.bell-sw.com/vm/21.0.0.2/python-installable-openjdk11-
21.0.0.2-linux-x64-musl.jar,nik,https://download.bell-
sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-openjdk11-21.0.0.2-linux-x64-
musl.tar.gz
ruby,https://download.bell-sw.com/vm/21.0.0.2/ruby-installable-openjdk11-
21.0.0.2-linux-amd64.jar,nik,https://download.bell-
sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-openjdk11-21.0.0.2-linux-
amd64.tar.gz
python,https://download.bell-sw.com/vm/21.0.0.2/python-installable-openjdk11-
21.0.0.2-linux-amd64.jar,nik,https://download.bell-
sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-openjdk11-21.0.0.2-linux-
amd64.tar.gz
,,nik,https://download.bell-sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-
openjdk11-21.0.0.2-linux-aarch64-musl.tar.gz
python,https://download.bell-sw.com/vm/21.0.0.2/python-installable-openjdk11-
21.0.0.2-macos-amd64.jar,nik,https://download.bell-
sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-openjdk11-21.0.0.2-macos-
amd64.zip
```

```
ruby,https://download.bell-sw.com/vm/21.0.0.2/ruby-installable-openjdk11-  
21.0.0.2-macos-amd64.jar,nik,https://download.bell-  
sw.com/vm/21.0.0.2/{COMPANY_NAME}-liberica-vm-openjdk11-21.0.0.2-macos-  
amd64.zip
```

17. Discover the nested LibERICA NIK component set only - getting a text mode response

In this example, we use the `components=ruby,python` request parameter to have only the nested NIK components in the text response.

In this case, no property of the parent base NIK components is included in the text response, except for the `components` property array. And if certain base NIK components do not contain the aforementioned nested NIK components, these base NIK components are excluded from the text response.

The JSON response is converted into the text response by applying the following rules to this JSON:

- All items in the text response are comma-separated
- The order of items in the text response corresponds to the order of values in the `fields` request parameter
- For each NIK component
 - No NIK component property names are printed, only NIK component property values
 - The only NIK component property values printed are those whose property names are mentioned in the `fields` request parameter
 - If the NIK component does not have the appropriate property, the value of this property is replaced by empty string
- If the `components` property of the base NIK component is an empty array (this base NIK component has no nested NIK components), this empty array is replaced by the sequence of empty strings equal to the number of property names in the `fields` request parameter (except for the `components` property name itself)
- If the base NIK component has no nested NIK components, it is displayed in the text response as a single line
- if the base NIK component contains nested NIK components (the `components` property of this base NIK component is a non-empty array), it is presented in the text response as several lines, equal to the number of the nested NIK components

 Important:

If the `components` request parameter includes any nested NIK component, the `fields` request parameter will include the components value *implicitly* as its last value.

Please compare the two **URLs** below: they both return the same result.

URL: <https://api.bell-sw.com/v1/nik/releases?components=ruby.python&fields=component,downloadUrl&output=text> <https://api.bell-sw.com/v1/nik/releases?components=ruby.python&fields=component,downloadUrl,components&output=text>

Text response example:

```
ruby,https://download.bell-sw.com/vm/21.0.0.2/ruby-installable-openjdk11-21.0.0.2-linux-x64-musl.jar
python,https://download.bell-sw.com/vm/21.0.0.2/python-installable-openjdk11-21.0.0.2-linux-x64-musl.jar
ruby,https://download.bell-sw.com/vm/21.0.0.2/ruby-installable-openjdk11-21.0.0.2-linux-amd64.jar
python,https://download.bell-sw.com/vm/21.0.0.2/python-installable-openjdk11-21.0.0.2-linux-amd64.jar
python,https://download.bell-sw.com/vm/21.0.0.2/python-installable-openjdk11-21.0.0.2-macos-amd64.jar
ruby,https://download.bell-sw.com/vm/21.0.0.2/ruby-installable-openjdk11-21.0.0.2-macos-amd64.jar
```

Bash example:

```
curl https://api.bell-sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl&output=text
curl https://api.bell-sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl,components&output=text
```

```
Invoke-WebRequest https://api.bell-sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl&output=text
Invoke-WebRequest https://api.bell-sw.com/v1/nik/releases?components=ruby,python&fields=component,downloadUrl,components&output=text
```



BellSoft Product
Discovery API

be//soft