

Paternoster ERP - Module

Grundlagen:

Paternoster ERP basiert auf das modulbasierte AF-Prototype Inhouse Framework.

Die Module werden in 3 Kategorien unterteilt, auch wenn das Framework intern keine Unterscheidungen macht.

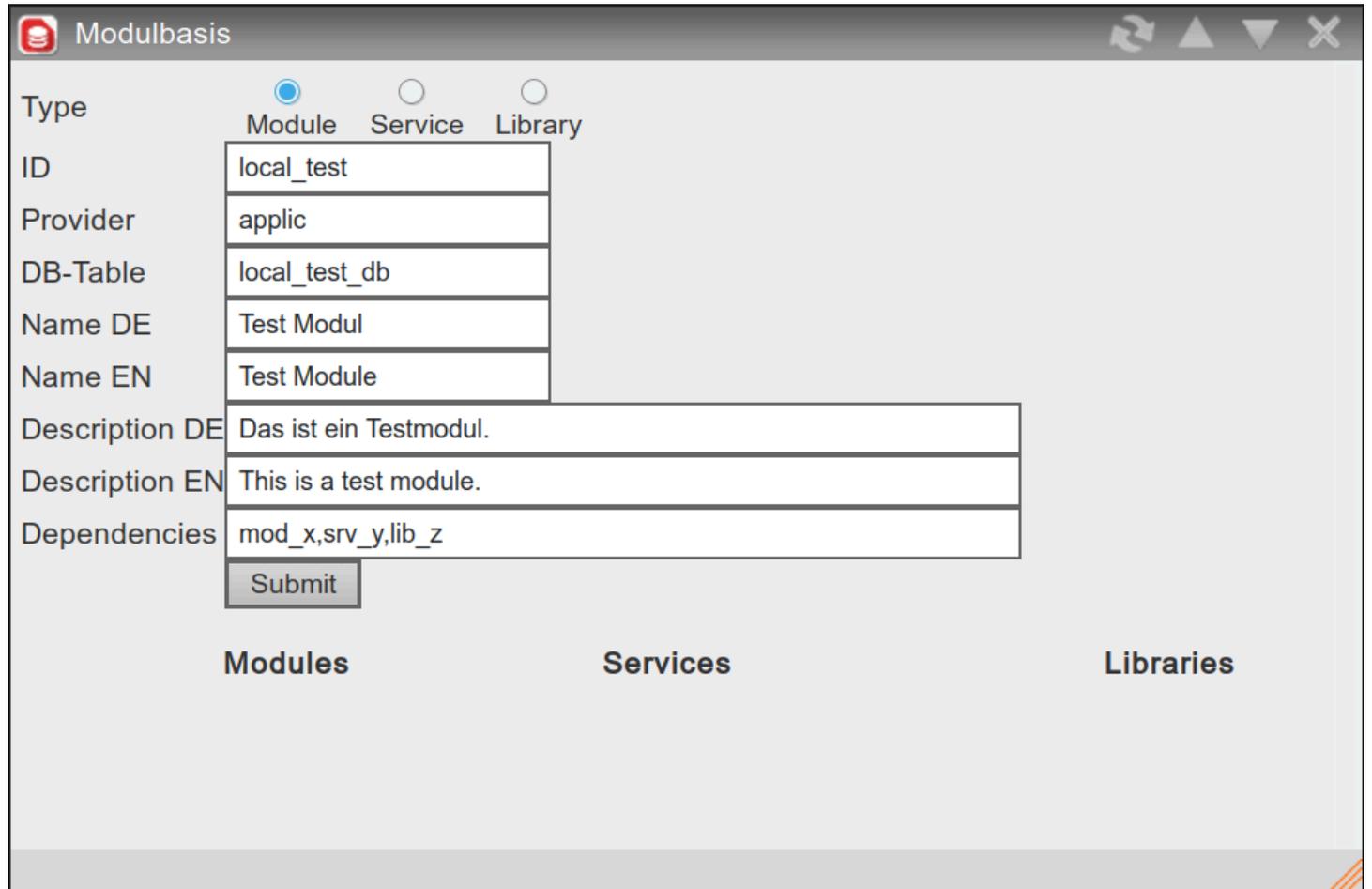
- **[lib] Libraries/Bibliotheken:** Diese Module stellen Funktionen bereit, die es Anwendungsentwicklern ermöglicht, vorhandenen Code für allgemeine Aufgaben wiederzuverwenden. z. B. ziplib, smtp lib, pdf lib, chart lib etc.
- **[srv] Services/Dienste:** Diese Module übernehmen Aufgaben, auf die Anwendungsentwickler zurückgreifen können. z. B. Beschaffung von Daten aus der Datenbank, Aufbau von Menüs, Versenden von E-Mails etc.
- **[mod] Modules/Module:** Die eigentlichen Module sind für den Funktionsumfang der Applikation verantwortlich. z. B. Artikelverwaltung, Kundenverwaltung, Anlagenverwaltung etc.

Die Steuerung dieser Module wird bei größeren Applikationen schnell unübersichtlich, da Dienste auch auf andere Dienste zurückgreifen können, die wiederum auf andere Dienste oder Bibliotheken zurückgreifen usw.

Aus diesem Grund gibt es eine Abstraktionsschicht, welche für den reibungslosen Ablauf aller Module zuständig ist. Dies setzt voraus, dass alle Module eine Installationsanleitung benötigen, die Anweisungen zur Registrierung beinhalten.

Modulbasis:

Mit diesem Entwicklungstool lassen sich Module bzw. Softwarepakete erstellen. Diese Module sind Basismodule, die keinerlei Funktionalität beinhalten.

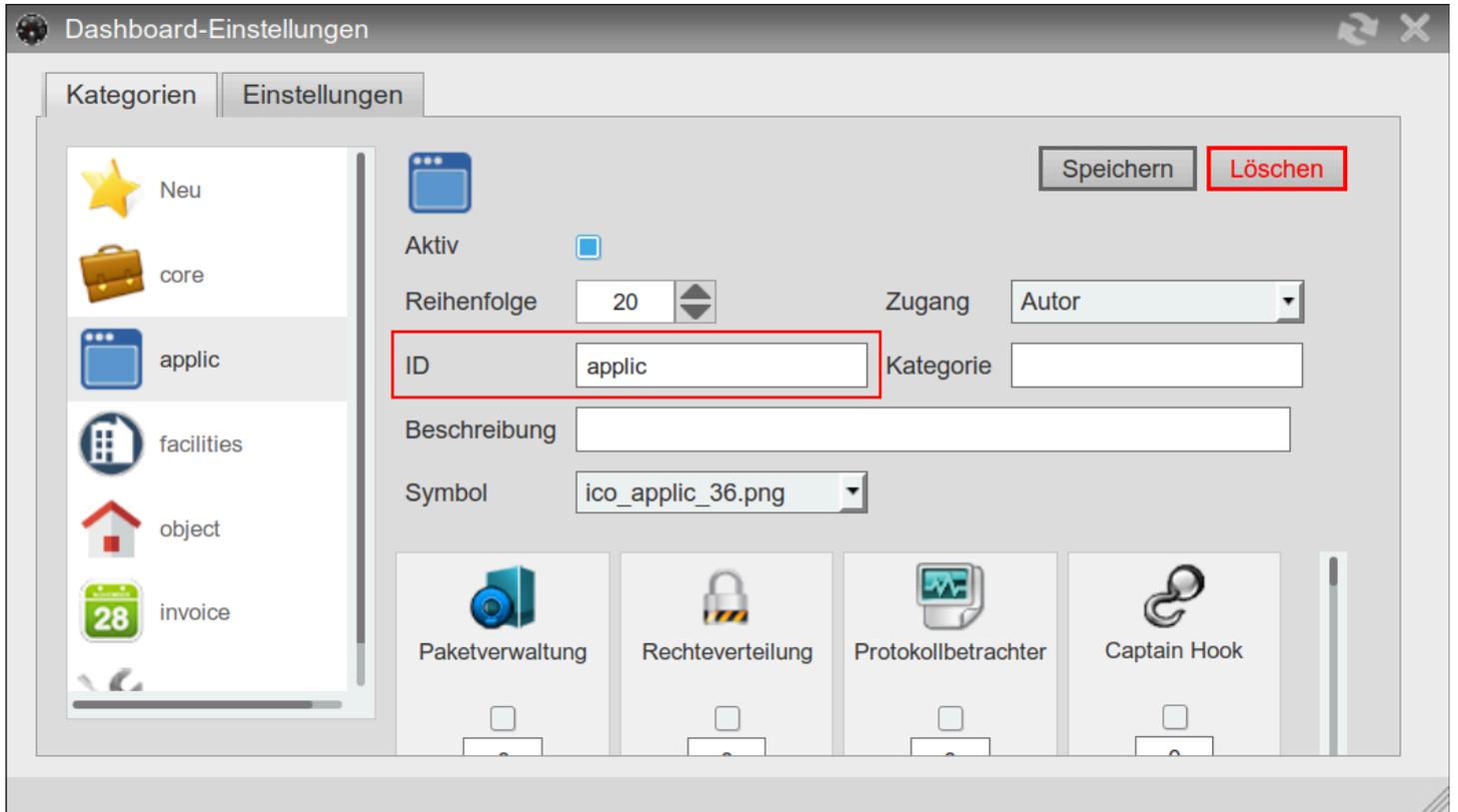


The screenshot shows the 'Modulbasis' application window. At the top, there are three radio buttons for selecting the type of module: 'Module' (selected), 'Service', and 'Library'. Below this, a form contains the following fields:

Type	<input checked="" type="radio"/> Module <input type="radio"/> Service <input type="radio"/> Library
ID	local_test
Provider	applic
DB-Table	local_test_db
Name DE	Test Modul
Name EN	Test Module
Description DE	Das ist ein Testmodul.
Description EN	This is a test module.
Dependencies	mod_x,svr_y,lib_z

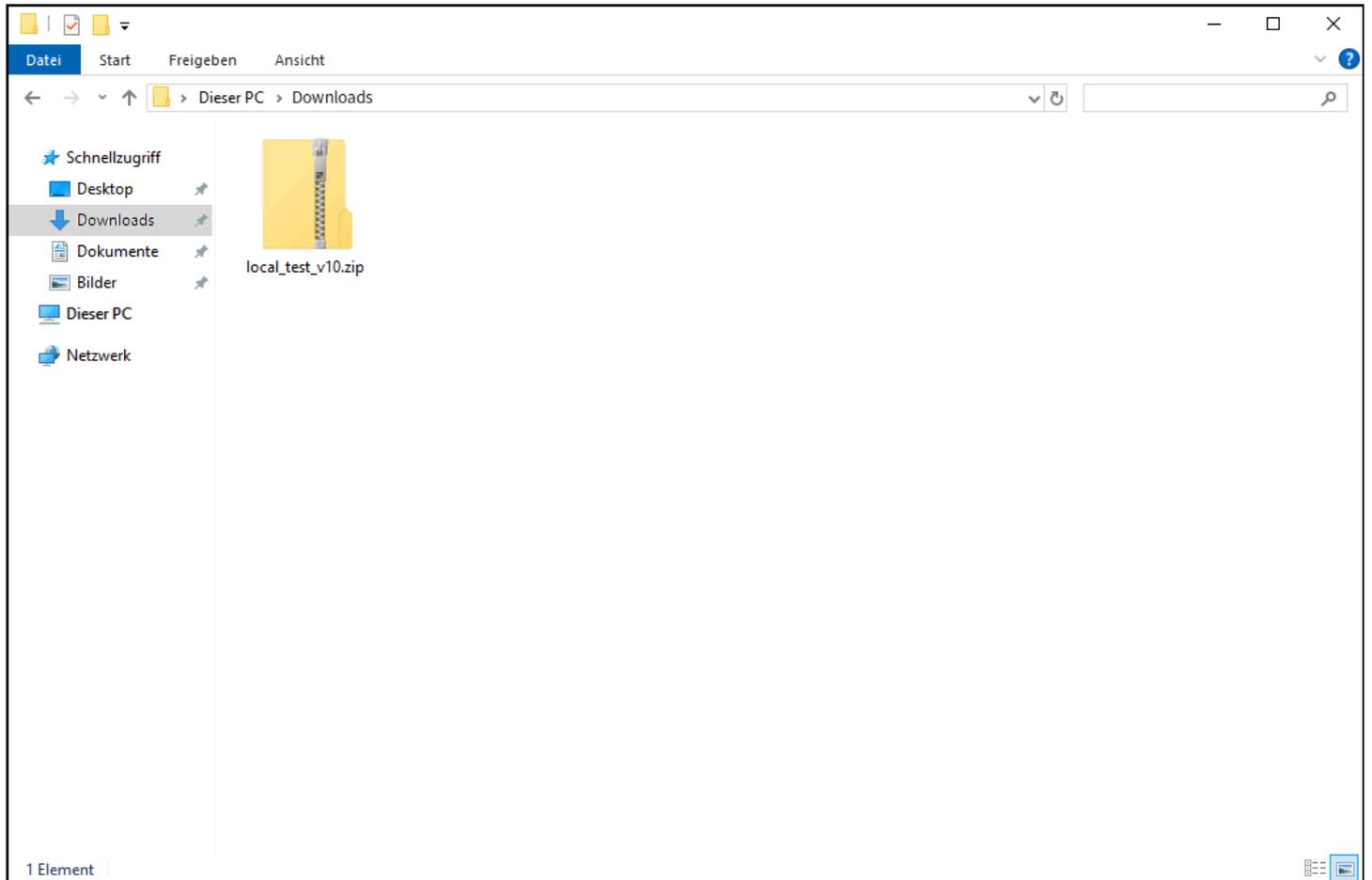
Below the form is a 'Submit' button. At the bottom of the window, there are three tabs: 'Modules', 'Services', and 'Libraries', with 'Modules' being the active tab.

- **Type:** Hier wird die Kategorie bzw. Modulart festgelegt.
- **ID:** Diese ID dient als Identifikator und muss eindeutig sein. Für die lokale Entwicklung wird empfohlen den „local“ Präfix zu verwenden um nicht mit anderen Modulen zu kollidieren.
- **Provider:** Der Provider ist ein Dienst, der Module mit Menükategorien verknüpft. So können Module bei der Installation z. B. mit der Menükategorie Stammdaten, Anwendungen oder Anlagenverwaltung verknüpft werden. Jede Menükategorie hat eine ID, die mit dieser Angabe übereinstimmen muss. Die ID der Menükategorien kann im Modul „Dashboard“ eingesehen werden. Diese Angabe ist für Bibliotheken oder Dienste nicht notwendig.

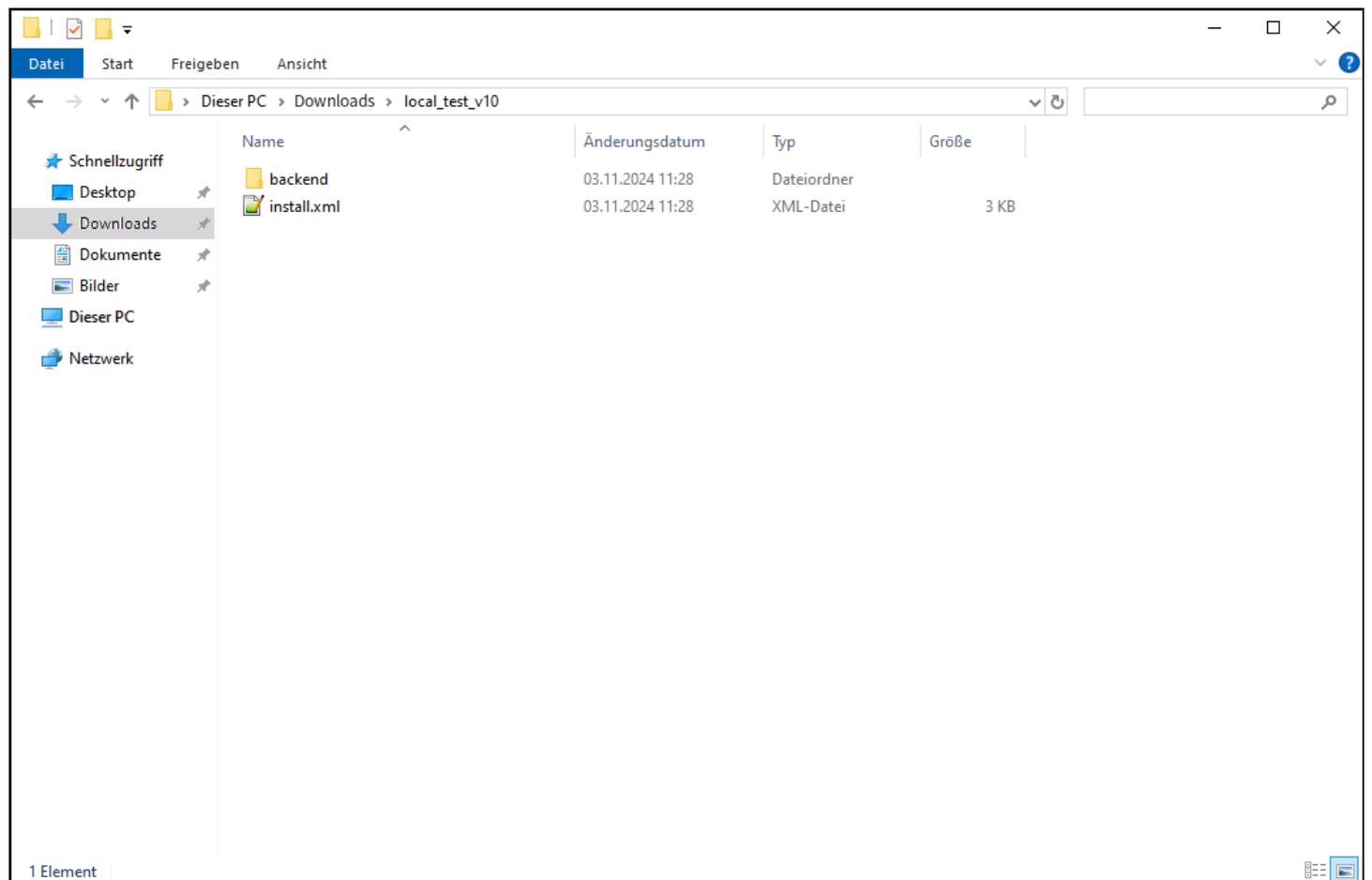


- **DB-Table:** Hier wird der Tabellename festgelegt, wenn das Modul mit Datenbankanbindung entwickelt wird. Für die lokale Entwicklung wird empfohlen den „local“ Präfix zu verwenden um nicht mit anderen Tabellen zu kollidieren.
- **Name DE:** Der Name des Moduls in Deutsch.
- **Name EN:** Der Name des Moduls in Englisch.
- **Description DE:** Eine Beschreibung des Moduls in Deutsch.
- **Description EN:** Eine Beschreibung des Moduls in Englisch.
- **Dependencies:** Mit Komma separierte Modul IDs, die für das zu entwickelnde Modul benötigt werden und als Abhängigkeit registriert werden.

Nach Erstellung des Moduls, wird das fertige Softwarepaket unter der Modulart aufgelistet und kann heruntergeladen werden.



Im entpackten Verzeichnis befindet sich das „backend“ Unterverzeichnis und die „install.xml“ Datei.



- **backend:** In diesem Unterverzeichnis befinden sich die Moduldateien für das BackEnd.
- **install.xml:** Diese Datei beinhaltet die Installationsanweisungen des Moduls.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <modules>
    <module>
      <data>
        <id>local_test</id>
        <name>Test Module</name>
        <version>1.0.0</version>
        <client>1.0.0</client>
        <date>2024-11-03</date>
        <method>install</method>
        <license>basic-erp</license>
        <copyright>2024 - # by basic-erp. All rights reserved.</copyright>
        <dev>basic-erp</dev>
        <devEmail>info@basic-erp.com</devEmail>
        <devSite>http://www.basic-erp.com/</devSite>
        <description>This is a test module.</description>
        <dependencies>mod_x, srv_y, lib_z</dependencies>
      </data>
      <settings>
        <terms>1</terms>
        <homePageBack>1</homePageBack>
        <homePageFront>0</homePageFront>
        <permaBack>1</permaBack>
        <permaFront>0</permaFront>
        <controllerBack>local_test/fctrl.php</controllerBack>
        <controllerFront></controllerFront>
        <langPath>local_test/lang/trm/</langPath>
        <iconPath>local_test/data/ico/</iconPath>
      </settings>
      <accesses>
        <access>local_test::Access Test Module::4</access>
        <access>local_test_insert::Insert Test Module::4</access>
        <access>local_test_update::Update Test Module::4</access>
        <access>local_test_delete::Delete Test Module::3</access>
      </accesses>
      <backendDir>backend/local_test</backendDir>
      <frontendDir></frontendDir>
    </module>
  </modules>
  <providers>
    <provider>
      <id>local_test</id>
      <passive>0</passive>
      <name></name>
      <description></description>
      <tooltip></tooltip>
      <ajaxBack>modules/local_test/res/ui/run.php</ajaxBack>
      <ajaxFront></ajaxFront>
      <langPath>modules/local_test/lang/trm/</langPath>
      <iconPath>modules/local_test/data/ico/</iconPath>
      <menuId>applic</menuId>
      <winTop>50</winTop>
      <winLeft>50</winLeft>
      <winWidth>420</winWidth>
      <winHeight>360</winHeight>
      <winResize>1</winResize>
      <winDebug>0</winDebug>
    </provider>
  </providers>
  <imports>
    <import>srv::local_test::modules/local_test/srv/class_loc.php</import>
  </imports>
  <database>

```

```

    <sql>
<![CDATA[
        CREATE TABLE IF NOT EXISTS `#__local_test_db` (
            `id_loc` INT(11) NOT NULL AUTO_INCREMENT,
            `loc_cro` DATETIME NOT NULL,
            `loc_crb` INT(6) NOT NULL,
            `loc_upo` DATETIME NOT NULL DEFAULT '0000-00-00 00:00:00',
            `loc_upb` INT(6) NOT NULL DEFAULT '0',
            `loc_lob` INT(6) NOT NULL DEFAULT '0',
            `loc_los` INT(32) NOT NULL DEFAULT '0',
            PRIMARY KEY (`id_loc`)
        ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_german2_ci AUTO_INCREMENT=1;
]]>
    </sql>
</database>
<uninstall>
    <database>
        <sql>DROP TABLE `#__local_test_db`</sql>
    </database>
</uninstall>
</root>

```

- **root/modules:** In diesem Element werden alle Module des Softwarepakets eingetragen, wobei jedes Modul in einem eigenen „module“ Element untergebracht ist. Auch wenn der Installationsdienst mehrere Module verarbeiten kann, wird empfohlen, dass jedes Softwarepaket nur ein Modul enthält.
- **root/modules/module/data:** Dieses Element enthält die Meta Daten des Moduls.
 - **id:** Die eindeutige ID des Moduls.
 - **name:** Der Name des Moduls.
 - **version:** Die Version des Moduls.
 - **client:** Die Mindestversion des Frameworks die das Modul benötigt.
 - **date:** Das Release Datum des Moduls.
 - **method:** Die Installationsart.
 - **license:** Der Lizenzinhaber des Moduls.
 - **copyright:** Die Urheberrechtinformationen.
 - **dev:** Name des Entwicklers.
 - **devEmail:** Kontakt E-Mail Adresse des Entwicklers.
 - **devSite:** Die Website / Landing Page des Entwicklers.
 - **description:** Die Beschreibung des Moduls.
 - **dependencies:** Die Abhängigkeiten die das Modul benötigt.
- **root/modules/module/settings:** Dieses Element enthält Einstellungspunkte für den Installationsdienst.
 - **terms:** Hier wird festgelegt, ob der Modul Name und die Modulbeschreibung in der Modul Datenbanktabelle eingetragen werden soll oder nicht.
 - **homePageBack:** Hier wird festgelegt, ob der Front Controller auf der Startseite des BackEnds aufgerufen werden kann.
 - **homePageFront:** Hier wird festgelegt, ob der Front Controller auf der Startseite des FrontEnds aufgerufen werden kann.
 - **permaBack:** Hier wird festgelegt, ob der Front Controller auf der Startseite des BackEnds aufgerufen wird, unabhängig ob dieser beansprucht wird.
 - **permaFront:** Hier wird festgelegt, ob der Front Controller auf der Startseite des FrontEnds aufgerufen wird, unabhängig ob dieser beansprucht wird.
 - **controllerBack:** Der Pfad zum Front Controller für das BackEnd ab dem Modul Verzeichnis.
 - **controllerFront:** Der Pfad zum Front Controller für das FrontEnd ab dem Modul Verzeichnis.
 - **langPath:** Der Pfad zum Sprachdateienordner ab dem Modul Verzeichnis.
 - **iconPath:** Der Pfad zum Symbol Ordner ab dem Modul Verzeichnis.

- **root/modules/module/accesses:** Dieses Element enthält die Zugangsberechtigungen.
 - **access:** Der Aufbau dieser Elemente hat 3 Anweisungen, die mit zwei Doppelpunkten getrennt sind und beginnt mit dem Berechtigungsschlüssel, der frei definierbar ist. Es wird empfohlen die ID des Moduls als Präfix zu benutzen, um nicht mit anderen Berechtigungsschlüsseln zu kollidieren. Der einzige Schlüssel, der vom System erkennbar ist, ist der Schlüssel, der die gleiche Bezeichnung wie die Modul ID hat. Dieser Schlüssel wird abgefragt, um zu entscheiden ob ein Modul für den Anwender zugänglich ist. Die zweite Angabe ist eine Beschreibung des Berechtigungsschlüssels. Bei der dritten Angabe wird die ID der Benutzergruppe angegeben, ab welcher die Berechtigung gegeben werden soll.
- **root/modules/module/backendDir:** Dieses Element enthält den Pfad zum Modul Ordner im Softwarepaket für das BackEnd.
- **root/modules/module/frontendDir:** Dieses Element enthält den Pfad zum Modul Ordner im Softwarepaket für das FrontEnd.
- **root/providers:** In diesem Element werden alle GUI Konfigurationen für den Modulprovider Dienst eingetragen. Wobei jedes Modul in einem eigenen „provider“ Element untergebracht ist und die Provider ID mit der Modul ID übereinstimmen muss.
- **root/providers/provider:**
 - **id:** Provider ID muss mit der Modul ID übereinstimmen.
 - **passive:** Hier wird festgelegt, ob das Modul vom System aufgebaut wird oder durch Eigenregie. Z. B. PopUp Fenster.
 - **name:** Standardmäßig wird der Name aus der Sprachdatei entnommen, sollte ein Name eingetragen sein, wird dieser bevorzugt.
 - **description:** Standardmäßig wird die Beschreibung aus der Sprachdatei entnommen, sollte eine Beschreibung eingetragen sein, wird diese bevorzugt.
 - **tooltip:** Standardmäßig wird der Kontext aus der Sprachdatei entnommen, sollte ein Kontext eingetragen sein, wird dieser bevorzugt.
 - **ajaxBack:** Pfad zur lauffähigen Datei für das BackEnd, die nach dem Aufbau des Moduls aufgerufen wird, beginnend mit dem Modul Sammelordner des BackEnds.
 - **ajaxFront:** Pfad zur lauffähigen Datei für das FrontEnd, die nach dem Aufbau des Moduls aufgerufen wird, beginnend mit dem Modul Sammelordner des FrontEnds.
 - **langPath:** Der Pfad zum Sprachdateienordner, kann von den Modulangaben abweichen, sollte aber wenn möglich derselbe sein.
 - **iconPath:** Der Pfad zum Symbol Ordner, kann von den Modulangaben abweichen, sollte aber wenn möglich derselbe sein.
 - **applic:** Die ID der Menükategorie, mit dem das Modul verknüpft werden soll.
 - **winTop:** Fixer Wert der für den Abstand des Modulfensters zum oberen Rand des Desktops verwendet werden soll.
 - **winLeft:** Fixer Wert der für den Abstand des Modulfensters zum linken Rand des Desktops verwendet werden soll.
 - **winWidth:** Fixer Wert für die Breite des Fensters.
 - **winHeight:** Fixer Wert für die Höhe des Fensters.
 - **winResize:** Hier wird festgelegt, ob das Fenster größenveränderbar ist.
 - **winDebug:** Hier wird festgelegt, ob Debug Informationen des Fensters angezeigt werden sollen, welche in der unteren Fensterleiste dargestellt werden.

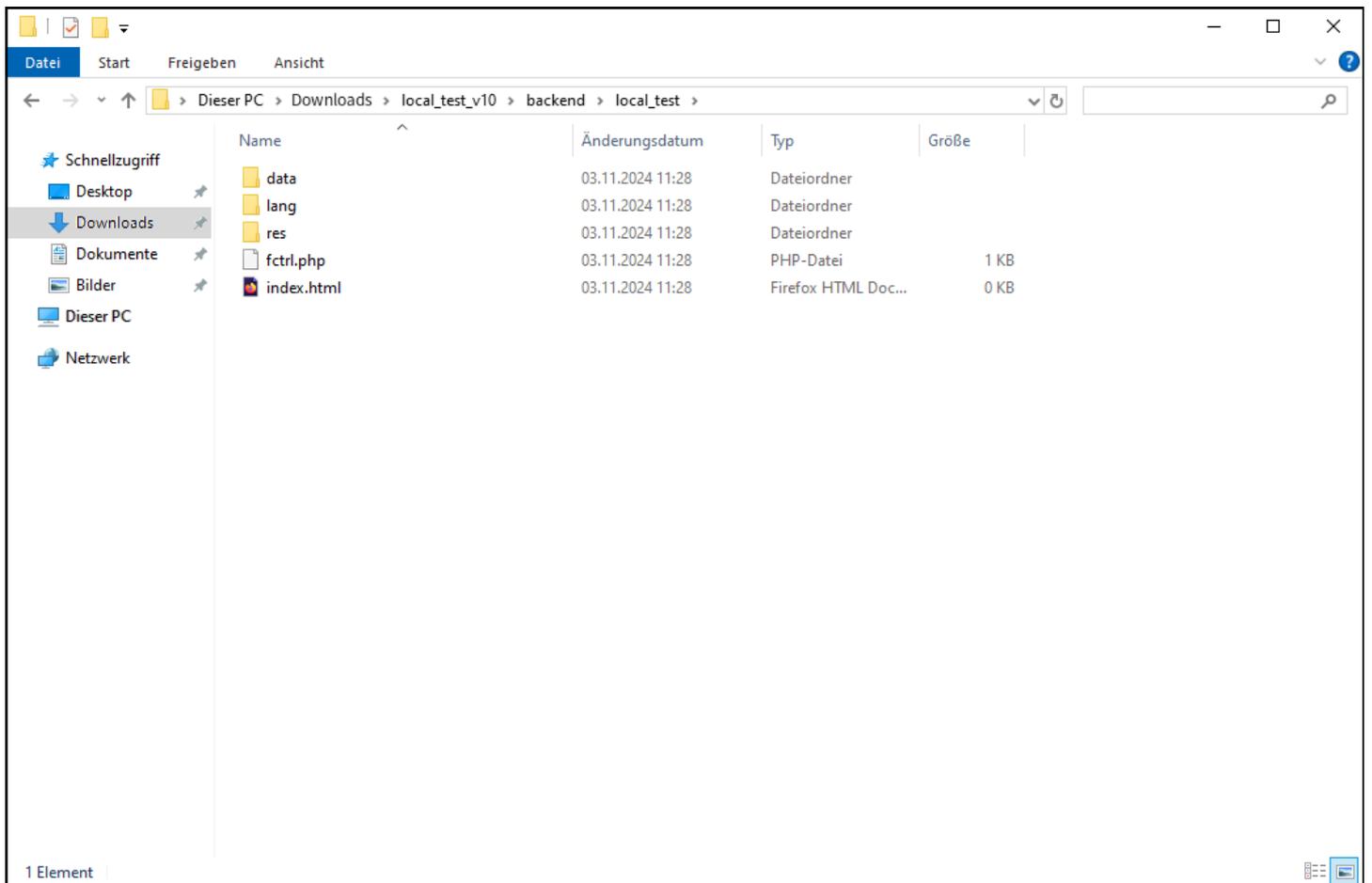
- **root/imports:** Dieses Element enthält die Importanweisungen, die das Modul zur Verfügung stellt, wobei jede Anweisung in einem eigenen „import“ Element untergebracht wird.
 - **import:** Der Aufbau dieser Elemente hat 3 Anweisungen, die mit zwei Doppelpunkten getrennt sind und beginnt mit der Importgruppe, die frei definierbar ist. Es wird empfohlen die ID des Moduls zu benutzen, um nicht mit anderen Importgruppen zu kollidieren. Die zweite Angabe ist der Importschlüssel. Bei der dritten Angabe wird der Pfad zur Importdatei angegeben, beginnend mit dem Modul Sammelordner.
 - **Globale Importgruppen:**
 - **request:** Importdateien für Request Dienste.
 - **string:** Importdateien für String Dienste.
 - **array:** Importdateien für Array Dienste.
 - **lang:** Importdateien für Sprachdateien.
 - **config:** Importdateien für Konfigurationsdienste.
 - **access:** Importdateien für Berechtigungsdienste.
 - **js:** Importdateien für JavaScript.
 - **lib:** Importdateien für Bibliotheken.
 - **dir:** Importdateien für Verzeichnis Dienste.
 - **file:** Importdateien für Datei Dienste.
 - **server:** Importdateien für serverbezogene Dienste.
 - **content:** Importdateien für Content Dienste.
 - **utils:** Importdateien für Utility Dienste.
 - **data:** Importdateien für CSS und JavaScript Registrierungsdateien.
 - **interface:** Importdateien für Interfaces.
 - **srv:** Importdateien für globale Dienste.
 - **mod:** Importdateien für globale Module.
- **root/database:** In diesem Element werden alle Datenbank Anweisungen eingetragen. Wobei jede Anweisung in einem eigenen „sql“ Element untergebracht ist. Die Reihenfolge ist zu berücksichtigen, wenn Foreign Keys verwendet werden, da die Anweisungen der Reihe nach aufgerufen werden. Auch sollten die Anweisungen in einem CDATA Block untergebracht sein, um Probleme beim Auslesen der XML Datei zu vermeiden.
- **root/database/sql:** Das SQL Statement sollte auf Funktionsfähigkeit geprüft sein. Wenn im Statement ein Datenbanktabellenname enthalten ist, kann ein „#__“ vorangestellt werden, der bei der Installation mit dem aktuellen Tabellenpräfix ausgetauscht wird.
- **root/uninstall:** In diesem Element werden alle Deinstallationsanweisungen eingetragen.
- **root/uninstall/database:** In diesem Element werden alle Datenbank Deinstallationsanweisungen eingetragen. Wobei jede Anweisung in einem eigenen „sql“ Element untergebracht ist.

Ordnerstruktur:

Im Unterverzeichnis „backend“ befinden sich die zu installierenden Modulordner, in diesem Beispiel lediglich der „local_test“ Ordner.

Die Ordnerstruktur eines Moduls ist nicht vorgegeben, sollte sich aber nach Möglichkeit an die Standards halten.

- **data:** Dieser Ordner enthält standardmäßig 4 Unterverzeichnisse.
 - **css:** Hier werden die CSS-Dateien des Moduls untergebracht.
 - **ico:** Hier befinden sich die Modulsymbole.
 - **img:** Hier befinden sich Bilder die das Modul verwendet.
 - **js:** Hier befinden sich die JavaScript Dateien.
- **lang:** Dieser Ordner enthält die Sprachdateien.
- **res:** Im Ressourcenordner befinden sich alle PHP Dateien die das Modul benötigt.
- **srv:** Optional kann ein Modul Dienste für sich benötigen oder für andere Module bereitstellen, die in dem Serviceordner untergebracht sind.



Front Controller:

Der FrontController wird beim Aufbau des Moduls aufgerufen und soll alle notwendigen CSS und JavaScript Dateien, die das Modul benötigt im System registrieren.

Das System ist über „\$system“ zugänglich.

```
$system -> addCssLink ( 'modules/local_test/data/css/local_test.css' ) ;  
$system -> addJsLink ( 'modules/local_test/data/js/local_test.js' ) ;
```

Sollte das Modul Dienste benötigen, muss eine entsprechende Importanweisung vorhanden sein, damit diese Dienste ebenfalls ihre benötigten Ressourcen registrieren können bzw. zur Verfügung stellen.

```
import ( 'data', 'lister,ribbon_menu,table', $system ) ;  
import ( 'js', 'confirm,invoice', $system ) ;
```

Repositoryum:

Softwarepakete können „on the fly“ installiert werden, die entsprechenden Installationsdienste benötigen lediglich eine URL zur Softwarepaketdatei.

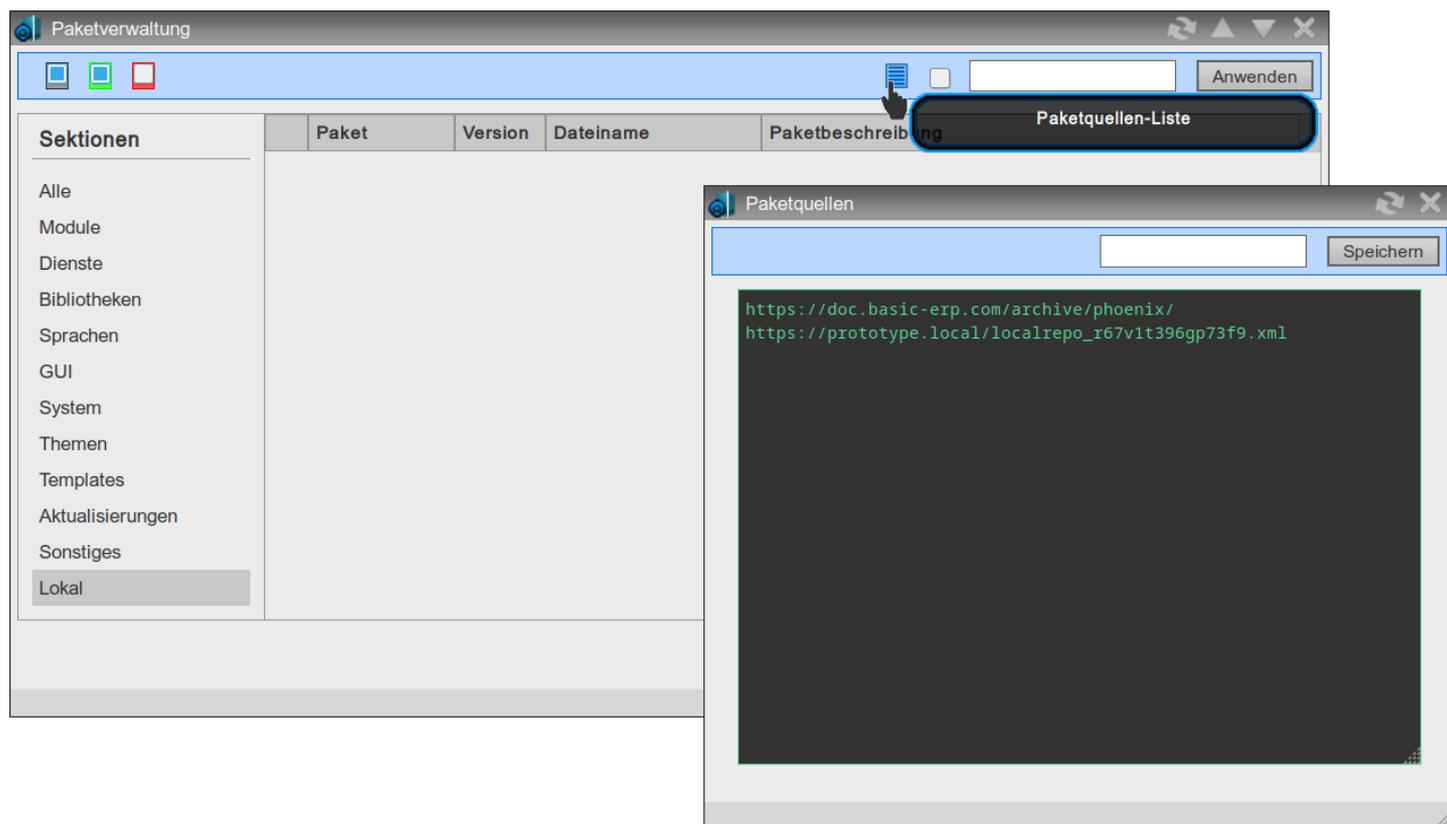
Dennoch wird empfohlen ein Repositoryum aufzustellen, in der alle Softwarepakete aufgelistet werden, um diese über die Paketverwaltung zu installieren.

Lokale Module werden nicht von basic-erp im globalen Repositoryum aufgenommen, eine entsprechende Infrastruktur ist diesbezüglich nicht vorhanden und ist auch nicht beabsichtigt.

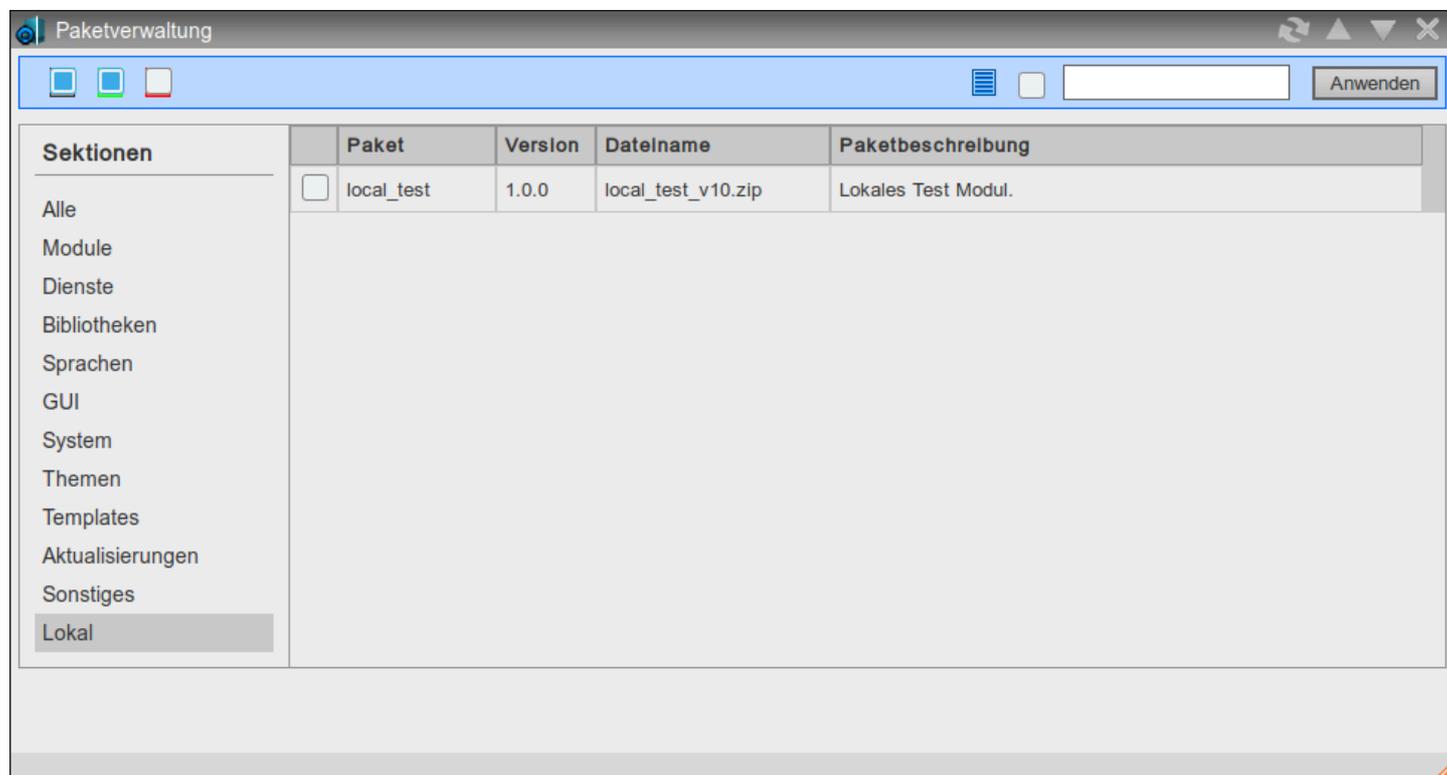
```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <package>
    <id_num>50001</id_num>
    <section>local</section>
    <dependent></dependent>
    <description>Lokales Test Modul.</description>
    <id>local_test</id>
    <version>1.0.0</version>
    <file>local_test_v10.zip</file>
    <path>http://prototype.local/localrepo/local_test_v10.zip</path>
  </package>
</root>
```

- **root:** Das Rotelement enthält alle Softwarepakete des Repositoryums, wobei jedes Paket in einem eigenen „package“ Element angegeben wird.
 - **root/package:**
 - **id_num:** Eine numerische, fortlaufende ID, die bei lokalen Repositoryums mit einem hohen Wert beginnen sollte, um nicht mit den IDs der globalen Repositoryums zu kollidieren.
 - **section:** Die Sektion, in welcher das Paket in der Paketverwaltung aufgelistet werden soll.
 - **dependent:** Die Paketabhängigkeiten für den Installationsdienst.
 - **description:** Die Beschreibung des Pakets, welche in der Paketverwaltung dargestellt werden soll.
 - **id:** Die Modul bzw. Paket ID.
 - **version:** Die Modul bzw. Paket Version.
 - **file:** Der Dateiname des Softwarepakets.
 - **path:** Die vollständige URL zur Datei.

In der Paketverwaltung muss das Repository registriert werden. Die URL Adresse jedes Repositoriums wird in einer eigenen Zeile angegeben.

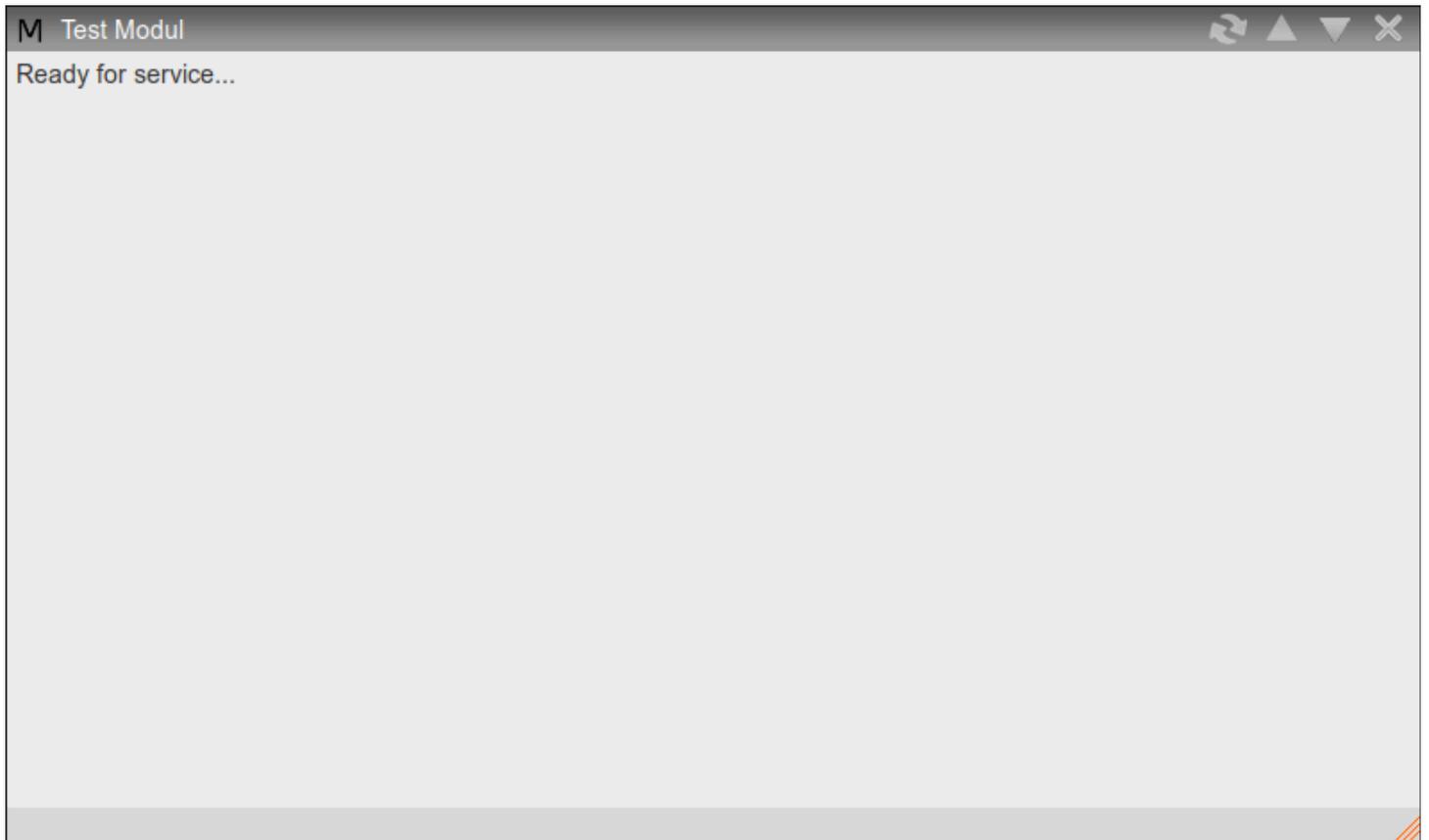


Nach dem Speichern der Quellen Liste, werden alle Pakete des hinzugefügten Repositoriums aufgelistet.



Hello World:

Nach der Installation des Moduls wird ein Standardtext ausgegeben.



Der Pfad zur lauffähigen Datei, die beim Aufbau des Moduls aufgerufen wird, wurde aus der Installationsdatei entnommen und ist in diesem Fall „modules/local_test/res/ui/run.php“

- **modules:** Der Backend Sammelordner für installierte Module.
- **local_test:** Der Modul Ordner.
- **res:** Der Modul Ressourcen Ordner.
- **ui:** Unterverzeichnis für die Anwendungsoberfläche.
- **run.php:** Lauffähige Datei.

Die lauffähige Datei erzeugt ein Model-View-Controller Entwurfsmuster (MVC), das lediglich einen Text ausgibt.

- **ctrl.php:** MVC Controller.
- **data.php:** MVC Model.
- **show.php:** MVC View.