

Entwickler-Handbuch

ene't

Inhaltsverzeichnis

1.	Voraussetzungen	3
1.1	Technische Voraussetzungen	4
1.2	Qualitätskriterien	5
2.	Authentifizierung	6
2.1	App registrieren	7
2.2	Aufruf Ihrer App	7
2.3	Authentifizierung beim ene't Navigator®	8
3.	Schnittstelle zur Prozesssteuerung	9
3.1	Prozesssteuerung	10
3.2	Schnittstelle	10
4.	BO4E-Repository	11
4.1	BOs in das Repository schreiben	12
4.2	BOs aus dem Repository lesen	13
5.	Legende	14

Einführung

Der ene't Navigator® ermöglicht die größtmögliche Flexibilität in der Abbildung von Geschäftsprozessen. Da er auf einem einheitlichen Standard basiert, können auf dem ene't Navigator® Anwendungen verschiedenster Hersteller kombiniert werden.

Schritt für Schritt leiten wir Sie in dieser Dokumentation durch die verschiedenen Schnittstellen, die Sie zur Anbindung einer App an den ene't Navigator® implementieren müssen.

Sollten Sie weitere Fragen haben, melden Sie sich gerne jederzeit bei uns.



Tim Steufmehl

Vertrieb, Partnermanager

Telefon: +49 (0)2433 52601-814 | E-Mail: vertrieb@enet.eu

1. Voraussetzungen

Der Navigator bietet als Web Plattform viele Freiheiten. Nichtsdestotrotz müssen Apps bestimmte Voraussetzungen erfüllen, um in den Navigator und somit auch in die Geschäftsprozesse eingebunden werden zu können.

1.1 Technische Voraussetzungen

■ Folgende Voraussetzungen müssen App-Anbieter erfüllen:

Bei der Anwendung muss es sich um eine Webanwendung handeln

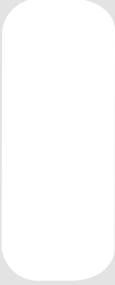
- läuft vollständig im Browser
- ist über das Internet erreichbar (ene't bietet auch Hosting-Dienstleistungen an. Sprechen Sie uns gerne an, wenn wir Sie hier unterstützen können.)
- ist ausschließlich über eine sichere Verbindung (HTTPS über TLS 1.2 oder höher) erreichbar.
- Grundsätzlich wird keine Programmiersprache vorgegeben. Empfohlen wird die Nutzung von Java bzw. Spring.
- Einbindung im Frameset muss erlaubt sein

1.2 Qualitätskriterien

■ Folgende Qualitätskriterien müssen erfüllt werden:

- Das App-Frontend muss mit validem HTML5, CSS3 und ggf. JavaScript implementiert werden. Die Nutzung von Flash, Silverlight, Java Applets usw. ist nicht erlaubt.
- Die App muss ausreichend getestet sein (Unit-Tests und e2e-Tests), um sicherzustellen, dass sie ordnungsgemäß funktioniert. Um Bugs schnellstmöglich korrigieren zu können, wird die Nutzung von Continuous Integration und Continuous Delivery empfohlen.
- Der Seitenaufbau muss in einer für Webanwendungen üblichen Geschwindigkeit erfolgen. Werden Hintergrundprozesse ausgeführt, so müssen diese asynchron ablaufen. Der Benutzer soll einen entsprechenden Wartebildschirm, ggf. inklusive Fortschrittsanzeige präsentiert bekommen. Sollen vollständige Apps im Hintergrund laufen (also ohne GUI), muss eine Service-App erstellt werden.
- Die Sicherheit der App muss vollständig vom App-Anbieter gewährleistet werden. Insbesondere betrifft dies
 - Vermeidung typischer Sicherheitsschwachstellen von Webanwendungen (z. B. OWASP Top 10)
 - Datenschutz, Datensicherheit, Datenintegrität, Transaktionssicherheit usw. müssen gewährleistet werden
 - Validierung von Eingaben (im Frontend und ggf. im Backend)
 - Logging von Fehlern, Monitoring des App-Verhaltens etc.

2. Authentifizierung



Authentifizierung und Autorisierung sind wichtige Bestandteile des Navigators. Damit Apps mit dem Navigator agieren können, müssen diese zunächst registriert werden. Mit der Registrierung erhalten Apps im Rahmen ihrer Berechtigungen die Möglichkeit, Informationen abzurufen und Aktionen durchzuführen.

2.1 App registrieren

■ Soll eine App in Prozesse eingebunden werden, muss diese am ene't Navigator® registriert werden. Bitte teilen Sie uns dafür folgende Informationen mit:

- Bezeichnung der App
- Die URL für den Einstieg in die App
- Fax- oder Handynummer (zur einmaligen sicheren Übermittlung des Passworts nach dem 2-Faktor-Prinzip)

Nach dem Erhalt dieser Informationen wird für die App eine eindeutige Client ID, ein Client Secret und eine Redirect URI generiert, die wir an Sie übermitteln. Diese Credentials werden in Ihrer Anwendung fest hinterlegt, damit sich Ihre App im ene't Navigator® authentifiziert und so am Prozessablauf teilnehmen kann.

2.2 Aufruf Ihrer App

■ Beim Aufruf Ihrer App werden diese drei Parameter übergeben, die von der App verarbeitet werden müssen:

- processInstancelId (im Pfad)
- taskId (im Pfad)
- accesscode (als Query Parameter)

Alle Parameter werden von der App für die weitere Kommunikation im ene't Navigator® benötigt.

Der Aufruf Ihrer App kann beispielsweise folgendermaßen aussehen:

```
https://www.ihre-domain.de/pfad/zu/ihrer/app/<processInstancelId>/  
<taskId>?access_code=<accesscode>&mandant_id=<mandantId>
```

Diese Daten werden im Kontext des Aufrufs für die weitere Kommunikation mit dem ene't Navigator® benötigt.

2.3 Authentifizierung beim ene't Navigator®

Mit Hilfe des Access Codes muss zunächst ein Access Token abgefragt werden:

```
GET <umgebung>/plattform-be/oauth/accesstoken?  
    process_instance_id=<processInstancelId>&  
    task_id=<taskId>&  
    code=<accesscode>&  
    grant_type=authorization_code&  
    redirect_uri=<redirectUri>
```

Bei diesem Aufruf muss zusätzlich ein Basic Authentication Header gesetzt werden. Den Inhalt bilden die Client ID und das Client Secret Base64-kodiert (nach dem Schema „Client-Id:Client-Secret“). Nachdem der Access Token abgefragt wurde, ist der Access Code nicht mehr gültig (die Abfrage des Tokens ist daher pro Access Code nur ein einziges Mal möglich). Der Access Token kann anschließend für weitere Anfragen innerhalb dieser Task-Instanz genutzt werden.

Mögliche Fehler:

HTTP-StatusCode	Mögliche Ursache
401	<ul style="list-style-type: none">• Es wurde kein BasicAuthenticationHeader übergeben• Es wurden falsche Credentials übergeben
404	<ul style="list-style-type: none">• Es wurde kein oder ein falscher Access Code übergeben

3. Schnittstelle zur Prozesssteuerung

Alle Apps, die auf dem Navigator eingebunden werden, müssen in der Lage sein mit der Prozesssteuerung zu kommunizieren. Für diese Kommunikation wird eine Schnittstelle zur Verfügung gestellt.

3.1 Prozesssteuerung

Die Prozesssteuerung dient dazu, die einzelnen Apps eines Prozesses nacheinander ausführen zu können. Damit ein Prozess fortgesetzt werden kann, muss eine App der Prozesssteuerung mitteilen, dass diese abgeschlossen ist. Die Authentifizierung erfolgt mit Hilfe des Access Tokens.

3.2 Schnittstelle

Wurde die App vollständig durchlaufen, muss diese der Prozesssteuerung mitteilen, dass ihre Arbeit abgeschlossen ist. Die Prozesssteuerung kann daraufhin die nächste App starten oder ggf. die Prozessinstanz beenden. Die App nutzt hierfür den „continue“-Webservice der Prozesssteuerung:

```
GET <umgebung> /prozesssteuerung/continue/<processInstanceId>/  
<taskId>?access_token=<accessToken>
```

Hat die Prozesssteuerung den Aufruf erfolgreich entgegengenommen, so antwortet sie mit dem HTTP-StatusCode 201.

Die Authentifizierung erfolgt mit Hilfe eines Access Token, daher muss dieser immer vorhanden und gültig sein.

Mögliche Fehler:

HTTP-StatusCode	Mögliche Ursache
401	<ul style="list-style-type: none">Die Authentifizierung konnte aufgrund eines fehlerhaften Access Tokens nicht durchgeführt werden.

4. BO4E-Repository

Das BO4E-Repository dient der Speicherung der Business Objects, die im Rahmen eines Prozesses anfallen. Dabei können Apps BOs schreiben, die von anderen Apps wiederum gelesen werden können. Somit ist ein App-übergreifender Austausch innerhalb einer Prozessinstanz möglich. Die Kommunikation mit dem BO4E-Repository erfolgt über REST-Webservices. Aufbau und Inhalt der BOs orientieren sich am BO4E-Standard (<https://www.bo4e.de/>). Um auf das BO4E-Repository zugreifen zu dürfen, muss immer der Access Token, der der App bei der Authentifizierung mitgeteilt wurde, übergeben werden.

Die vollständige Dokumentation der BO4E-Repository Schnittstellen finden Sie unter <https://ws.enet-navigator.de/docs/ui?url=/docs/swagger/bo4e-repository>.

4.1 BOs in das Repository schreiben

■ Um BOs zu schreiben, werden diese per POST an das Repository gesendet:

POST `<umgebung>/bo4e-repository/bo4e?access_token=<accessToken>`

Als Body wird hier ein Transfer-Objekt mit verschiedenen Meta-Daten sowie dem BO übergeben. Die Struktur sieht folgendermaßen aus:

```
{
  "prozessId": "<processInstanceId>",
  "prozessSchritt": "<taskId>",
  "mandantId": "<mandantId>",
  "boTyp": "<boType>",
  "versionStruktur": "1",
  "boContent": [<boContent>, ...]
}
```

Die benötigten Werte für „prozessId“, „prozessSchritt“ und „mandantId“ entnehmen Sie den Query-Parametern beim App-Aufruf, wie in [2.2 Aufruf Ihrer App](#) beschrieben. Beim „boTyp“ handelt es sich um den Typen des BOs, wie in der BO4E-Dokumentation beschrieben. Beim „boContent“ handelt es sich um eine Liste, in der die vollständige Struktur der konkreten BOs in JSON-Form hinterlegt werden.

Das Repository bestätigt den Erhalt des BOs mit dem HTTP-StatusCode 200, sofern die Übertragung erfolgreich war und Inhalt und Struktur des BOs dem Standard entsprechen.

4.2 BOs aus dem Repository lesen

■ Es gibt mehrere Möglichkeiten, BOs auszulesen:

Abrufen aller BOs des Prozesses, die einen bestimmten BO-Typ (`<boType>`) haben:

```
GET <umgebung>/bo4e-repository/bo4e/<processInstanceId>/<boType>
?access_token=<accessToken>
```

Antwort ist eine Liste mit allen BOs (inklusive Meta-Daten) des angefragten Typs, die im Rahmen des Prozesses erzeugt wurden.

Abrufen eines bestimmten BOs über die ID (`<boId>`) des BOs:

```
GET <umgebung>/bo4e-repository/bo4e/<processInstanceId>/
boId?boId=<boId>&access_token=<accessToken>
```

Als Antwort wird das entsprechende BO (inkl. Meta-Daten), sofern vorhanden, zurückgegeben. Abrufen eines bestimmten BOs über den Typ (`<boType>`) des BOs:

```
GET <umgebung>/bo4e-repository/singlebo4e/<processInstanceId>/
<boType>?access_token=<accessToken>
```

Als Antwort wird das BO (inkl. Meta-Daten) zurückgegeben. Dies ist nur möglich, falls nur ein BO mit dem angefragten Typ im Rahmen des Prozesses angelegt wurde.

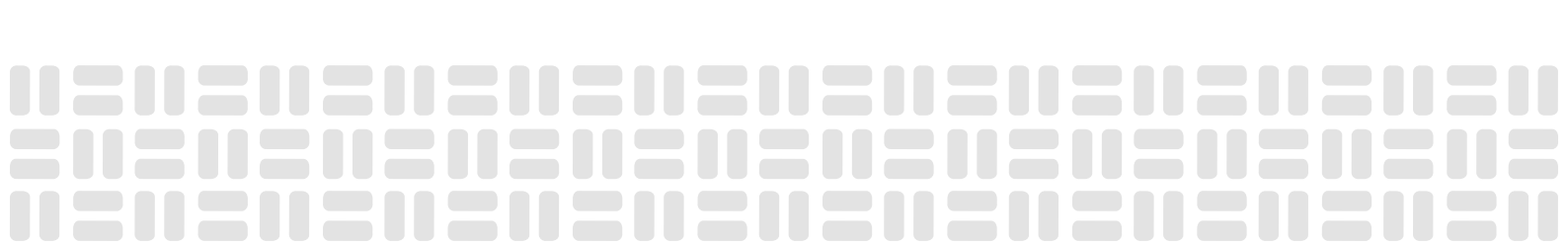
Mögliche Fehler:

HTTP-StatusCode	Mögliche Ursache
400	<ul style="list-style-type: none">• json (Body) konnte nicht dekodiert werden• Es sollte nur ein BO gelesen werden, es waren aber mehrere vorhanden• BO-Typ zum übermittelten <code><boType></code> konnte nicht ermittelt werden
401	<ul style="list-style-type: none">• Der <code><accessToken></code> ist nicht valide. Der Client ist nicht autorisiert das BO zu lesen oder zu schreiben

5. Legende

Blau hinterlegte Teile der Adressen sind Variablen

<code><accessToken></code>	dient zur Authentifizierung und muss bei jedem Aufruf mitgegeben werden
<code><boContent></code>	der Inhalt des BOs als JSON-Daten.
<code><boId></code>	ist die eindeutige ID eines BOs. Ein Zugriff per BO ID erfolgt in der Regel, wenn eine Referenz in einem BO aufgelöst werden soll.
<code><boType></code>	ist der Typ des BOs. Die vorhandenen Typen können den BO-Beschreibungen entnommen werden.
<code><processInstanceId></code>	ist die ID des Prozesses, die der App beim Aufruf in der URL mitgegeben wird.
<code><umgebung></code>	ist die entsprechende Umgebung <ul style="list-style-type: none">• www.enet-navigator.de (Produktivumgebung)• partner.enet-navigator.de (Partner-Umgebung für Entwicklungszwecke)



ene't

Weserstraße 9
41836 Hückelhoven
Telefon: +49 (0)2433 52601-0

Solarstraße 27
04860 Torgau
Telefon: +49 (0)3421 18391-0

info@enet.eu

www.enet.eu