



BGR Bohrpunktkarte Deutschland

**Borehole Profil Backend
Service**

Version 1.0.1

19.10.2016

Bundesanstalt für Geowissenschaften und Rohstoffe
Arbeitsbereich Geodatenmanagement

Stilleweg 2
30655 Hannover

Telefon: +49 (0)511 643 3433
E-Mail: BoreholeML@infogeo.de

1 Einleitung

Diese Spezifikation beschreibt die REST-Serverschnittstelle zur Darstellung der Bohrprofildaten und weiterer Detaildaten der Bohrpunktkarte Deutschland. Auch wenn diese Schnittstelle öffentlich zugänglich ist, gelten für die Nutzung der bereitgestellten Bohrdaten die jeweiligen Nutzungsbedingungen der Staatlichen Geologischen Dienste, welche die Bohrdaten in Form von BoreholeML anbieten. Die Bohrdaten der Länder werden über WFS 1.1.0 an die REST-Serverschnittstelle im Hintergrund angebunden. Die Schnittstelle fragt den jeweiligen Landes-WFS an und transformiert via XSLT das BoreholeML nach HTML. Eine Datenabgabe über BoreholeML über diese REST-Schnittstelle ist nicht möglich. Die Ansprechpartner finden zu den WFS-Diensten der Länder finden sie unter [infogeo.de](http://info.geo.de).

Der einfachste und empfohlene Weg die frei verfügbaren Bohrungsdaten aus der Bohrpunktkarte Deutschland einzubinden ist per WMS-Schnittstelle (v 1.3.0). Die Serviceadresse lautet:

<https://services.bgr.de/wms/geologie/gbl/>?

Auch bei WMS werden die Schichtdaten in Form von HTML + Javascript zum Client geschickt, der das auswerten muss. Getestet haben wir das mit folgenden WMS Clients ArcMap, ArcGIS Pro und QGIS.

Es wird keine Gewährleistung gegeben, dass diese Schnittstelle dauerhaft angeboten wird. Änderungen sind aufgrund von Anpassungen an der BoreholeML ebenfalls jederzeit möglich.

2 Borehole Profil Backend Service

Diese Komponente wird als Java Servlet Anwendung realisiert (REST). Sie übernimmt als zentrale Aufgabe die Abfrage und visuelle Aufbereitung der Detaildaten eines Bohrpunktes. Dazu werden die Anfragen serverseitig an die jeweiligen WFS weitergereicht.

Eine weitere Aufgabe dieser Java Servlet Anwendung ist die Bereitstellung XSLT Stylesheets, welches vom ArcGIS for Server verwendet wird, um die OGC WMS GetFeatureInfo Darstellung so zu übersteuern, dass die HTML Ansicht dargestellt wird.

Das Sequenzdiagramm in Abbildung 1 zeigt wie alle Komponenten in bestimmter Reihenfolge miteinander kommunizieren, um am Ende die Detaildatendarstellung als HTML-Dokument inkl. Javascriptanteil zum Client zu schicken.

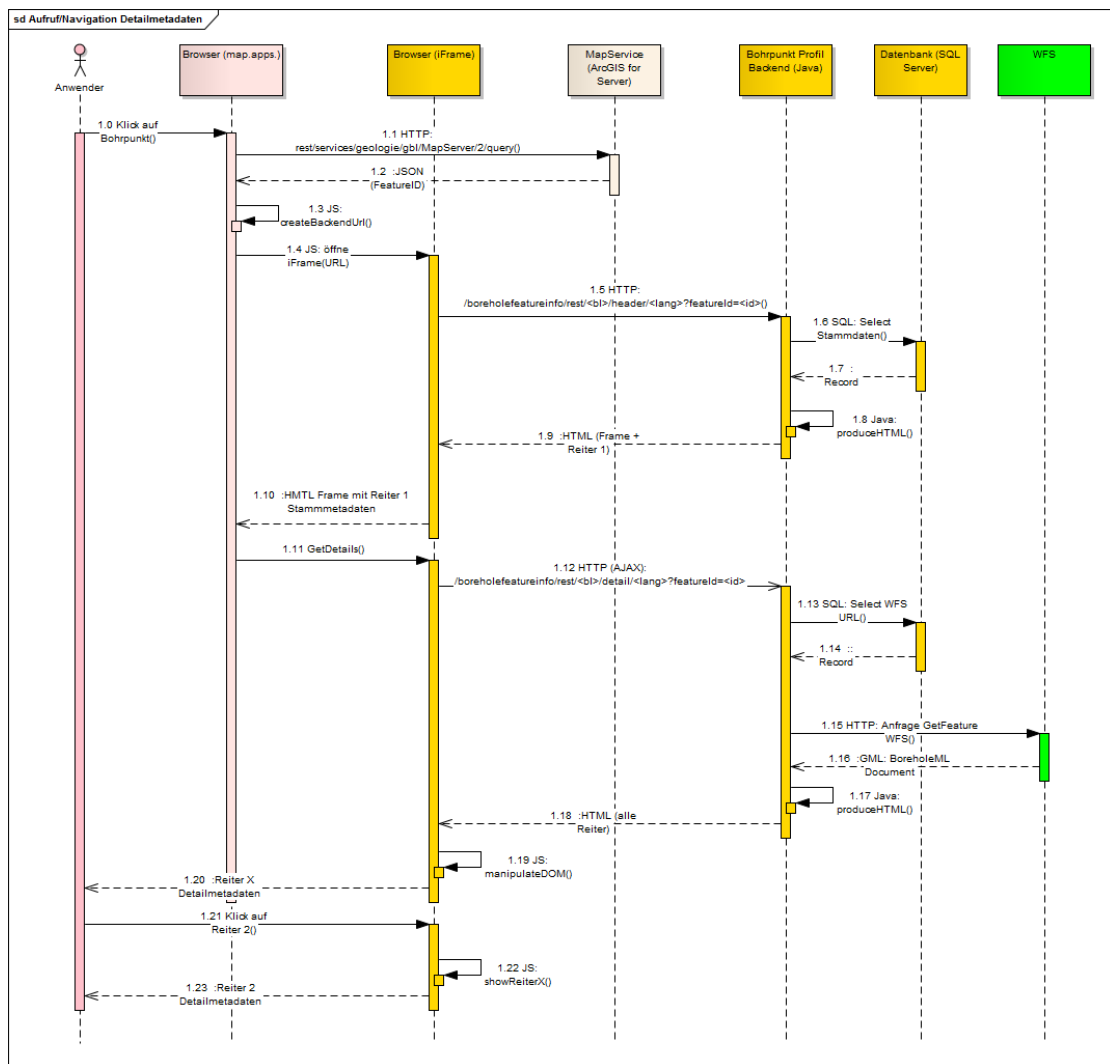


Abbildung 1: Sequenzdiagramm – ArcGIS Rest Interface

Parallel zur ArcGIS-Rest Schnittstelle wurde auch eine Lösung für WMS realisiert, die im Endeffekt einen „kaskadierenden WMS“ durch die Anbindung mit den BoreholeIML-WFS darstellt. Zur Realisierung wird ein XSLT beim FeatureInfo-Request mitgegeben, welches per Javascript-Code einen redirect vom Client auslöst, der gegen das Backend läuft.

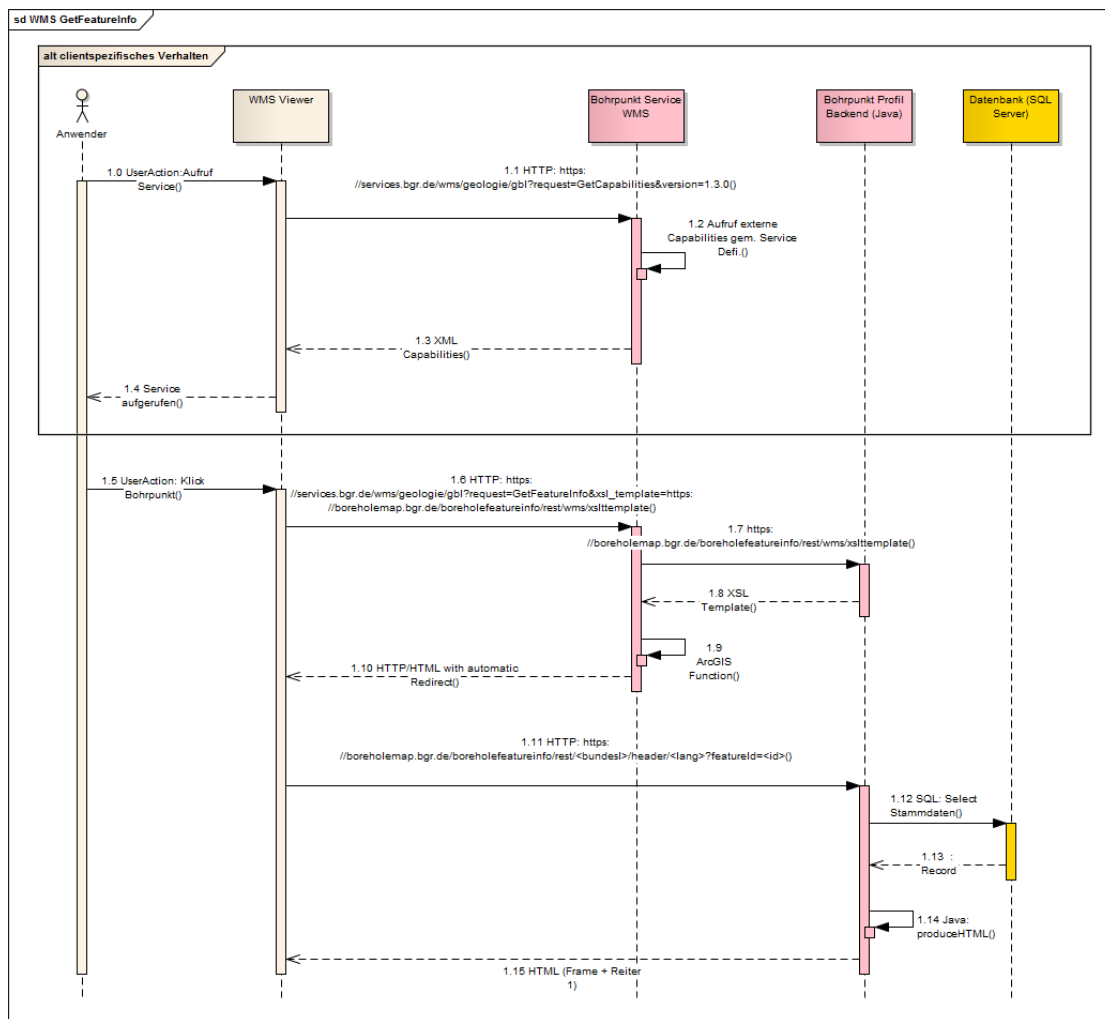


Abbildung 2: Sequenzdiagramm ArcGIS WMS Interface (unvollständig WFS-Request s. Abbildung2)

2.1 REST-Schnittstelle

Das Servlet verfügt über zwei REST-Schnittstellen. Eine Schnittstelle stellt das HTML Dokument der Detailmetadaten bereit. Die andere Schnittstelle dient dem Abruf des XSLT Stylesheet. Über eine UI-Schnittstelle verfügt diese Komponente nicht.

2.1.1 GET /rest/<bundes>/<segment>/<lang>?featureId=<id>

Die Basis-URL dieser REST-Schnittstelle lautet: <https://boreholemap.bgr.de/boreholefeatureinfo>

Beschreibung:

Stellt die Detailinformationen des entsprechenden Bohrpunktes dar.

Rückgabeformat/Content-Type:

- HTML: HTML-Dokument, wie in Anlage4_Mockup_Detaildaten.pdf beschrieben. Die Sprache richtet sich nach dem Wert des <lang> Parameters.

Pfad- und Query-Parameter

Parameter	Wert	Beschreibung	Datentyp
<bundesl>	z.B. sh, nw, etc...	Kürzel des jeweiligen Bundeslandes	String
<segment>	{header detail}	header: Liefert das komplette HTML Gerüst inkl. erstem Reiter mit Headerinformationen, die aus der Datenbank abgefragt werden. detail: liefert die Detailinformationen der übrigen Reiter, die vom jeweiligen WFS abgefragt werden. Externe Anwendungen sollten immer „header“ abfragen.	String
<lang>	{de en}	Sprache der Detailinformationen, falls unbekannte Sprache abgefragt wird, kommt als Default Deutsch zurück	String
<id>	z.B. DABO_10	GML-ID des Borehole Features. Die ID wird nicht als Pfad-Parameter übergeben, sondern als http Query Parameter, da IDs „/“ enthalten können.	String

Beispielaufwurf für die Headerinformationen (inkl. JS-Gerüst für die Reiter mit den Detaildaten):
[https:// boreholemap.bgr.de/boreholefeatureinfo/rest/nw/header/de?featureId=BDN_4127GE0027](https://boreholemap.bgr.de/boreholefeatureinfo/rest/nw/header/de?featureId=BDN_4127GE0027)

Beispielaufwurf für die Detailinformationen (nur sinnvoll, wenn zuvor die Header angefragt wurden):
[https:// boreholemap.bgr.de/boreholefeatureinfo/rest/nw/detail/de?featureId=BDN_4127GE0027](https://boreholemap.bgr.de/boreholefeatureinfo/rest/nw/detail/de?featureId=BDN_4127GE0027)

2.1.2 GET /rest/wms/xslttemplate

Die Basis-URL dieser REST-Schnittstelle lautet:
<https://boreholemap.bgr.de/boreholefeatureinfo>

Beschreibung:

Liefert das XSL Template zurück, welches den HTML-Redirect erzeugt.

Rückgabeformat/Content-Type:

- XSLT-Dokument
- Content-Type: application/xml

Beispielaufwurf zum holen des XSLT:

<https://boreholemap.bgr.de/boreholefeatureinfo/rest/wms/xslttemplate>

2.2 Technische Umsetzung

2.2.1 Webapplikation Borehole Profil Backend (Apache Tomcat)

Die Borehole Profil Backend wird als Java Webapplikation realisiert und in einem Apache Tomcat betrieben. Die Webapplikation ist so implementiert, dass eine Lastverteilung möglich ist (keine Sticky Session notwendig).

Verwendete Frameworks:

- Apache Camel (<http://camel.apache.org/>)
- Spring (<https://spring.io/>)

Struktur der Webapplikation:

- Dateiname: boreholefeatureinfo.war
Der Name ist Bestandteil der URL zum Aufruf der Komponente
- /images (Images der Bohrsäulengrafik)
- /META-INF
 - context.xml (Webkontext mit Referenz auf die JNDI Ressource in der server.xml)
- /WEB-INF
 - /classes
 - /sql
 - header.sql (Beschreibung siehe unten)
 - wfs.sql (Beschreibung siehe unten)
 - /wms
 - featureinfo_text_html.xml (Beschreibung siehe unten)
 - boreholeml2html.xml (Beschreibung siehe unten)
 - camel-context.xml (Camel-Komponentenkonfiguration der Webapplikation)
 - codelists.xml (Beschreibung siehe unten)
 - common.xml (Transformation der Codelisten)
 - header2html.xml (Beschreibung siehe unten)
 - profile.xml (Beschreibung siehe unten)
 - ui.properties (Beschreibung siehe unten)
 - /lib (verwendete Java Bibliotheken)
 - web.xml
- index.html (Standardseite)
- style.css (CSS Datei für die Darstellung der HTML Oberflächen)
- touch.css (CSS Datei für die Darstellung der HTML Oberflächen)

Komponenten:

Alle Komponenten dieser Schicht sowie der Informationsfluss zwischen ihnen werden über die zentrale Apache Camel Konfiguration definiert (unter WEB-INF\classes\camel-context.xml). Dieser Servlet-Stack verfügt über die folgende Struktur:

- **rest (REST Service):** Verarbeitet die eingehenden HTTP Anfragen, delegiert die Anfragen an die entsprechende Apache Camel Route und setzt auf dem Rückweg die richtigen HTTP Header.
- **Route getHeader (Zugriff auf Stammdaten):** Für den schnellen Zugriff auf die Stammdaten (erster Reiter mit Header Informationen) werden die Informationen direkt aus der Datenbank gezogen. Das Ergebnis der Datenbankabfrage wird mit JAXB in

- XML serialisiert und dann mit dem XSLT Stylesheet boreholeml2html in die Rückgabe übersetzt.
- **Route `getFeature` (WFS Aufruf):** Dieser Teil der Applikation kümmert sich um den Abruf der BoreholeML Objekte von den jeweiligen WFS. Für den WFS Zugriff müssen die jeweiligen Zugriffsinformationen zuerst aus der Datenbank gelesen werden (WFS-URL, ggf. Username und Passwort). Das vom WFS gelieferte XML Dokument wird mit dem XSLT Stylesheet boreholeml2html in die Rückgabe übersetzt.
 - **header2html.xsl (Erzeugung HTML Output erster Reiter):** Erzeugt aus Stammdateninformationen in der Datenbank, die über JAXB in XML serialisiert worden sind, das HTML Gerüst sowie die Informationen für den ersten Detailreiter mit den Header-Informationen.
 - **boreholeml2html.xsl (Erzeugung HTML Output):** Diese Komponente erzeugt aus dem vom WFS gelieferten Borehole GML-Objekt den HTML Output – für jeden zusätzlichen Reiter ein HTML Fragment. Dies beinhaltet die Erzeugung der Bohrsäulen-Grafik. Grundlage dafür ist der IT.NRW Code (siehe auch profile.xml).
 - **profile.xml (Bohrsäulendarstellung):** Diese Komponente enthält die Zuordnung von Bildern und Farbcodes zu den Gesteinsschichten, übernommen vom IT-NRW Code. Ausgangsdaten sind die Codelisten [RockNameList](#) und [ChronoStratigraphyList](#) der BoreholeML unter Infogeo.de
 - **codelists.xml (Erzeugung HTML Output):** Diese Komponente enthält die deutsche und englische Beschreibung für alle Codelisten aus der BoreholeML. Ausgangsdaten sind die [Codelisten zur BoreholeML](#) unter Infogeo.de
 - **ui.properties (Erzeugung HTML Output):** Diese Komponente enthält die deutschen und englischen Übersetzungen für alle statischen Oberflächenelemente.
 - **header.sql (Datenbankabfrage):** Diese Komponente enthält die SQL Abfrage, mit der die Header-Informationen aus der Stammdatentabelle abgefragt werden.
 - **wfs.sql (Datenbankabfrage):** Diese Komponente enthält die SQL Abfrage, mit der die WFS-URL und evtl. Nutzernamen/Passwörter für http Basic Authentication aus der SGD Tabelle abgefragt werden.
 - **Route `getTemplate`:** Liefert die XSLT Datei, mit der die GetFeatureInfo Response des ArcGIS Server WMS Schnittstelle übersetzt wird. Die Anfrage kommt direkt vom ArcGIS Server. Es muss dafür gesorgt werden, dass die Capabilities des Dienstes entsprechend übersteuert werden, so dass ein Standardclient auch den entsprechenden Parameter XSL_TEMPLATE in der GetFeatureInfo Anfrage an den ArcGIS Server übergibt. Erzeugt wird ein entsprechender Redirect auf die getHeader Route der Detailansicht.
 - **featureinfo_text_html.xsl (GetFeatureInfo Transformation):** Diese Komponente transformiert eine Antwort vom ArcGIS Server und konstruiert die URL auf die Detailansicht.

2.2.2 Web Feature Service

Für den Abruf aller Bohrinformationen wird der entsprechende WFS des Bundeslandes abgefragt. Ist der WFS über HTTP Basic abgesichert, werden die entsprechenden Credentials (Username und Passwort) mitgeschickt.

Die Komponenten stellen an die verschiedenen WFS-GetFeature Anfragen in der folgenden Art und Weise. Die Anfragen sind identisch für alle WFS:

- SERVICE=WFS
- REQUEST=GetFeature
- VERSION=1.1.0
- outputFormat=text/xml;subtype=gml/3.2.1

- featureID=<GML ID des Bohrloch Features>

Das Rückgabeformat orientiert sich an der BoreholeML Spezifikation in der Version 3.0.1. (Die Spezifikation ist online unter der URL: http://www.infogeo.de/home/boreholeML/current_version/index_html?lang=1).

3 Borehole MapService/WMS

3.1 Schnittstelle

Die Anforderungen an diese Schnittstelle bzw. die bereitgestellten Kartenebenen sind im Fachkonzept 'fachkonzept MapService bohrpunktkarteDE.docx' näher beschrieben.

3.1.1 ArcGIS Rest API (MapService)

Die Basis-URL dieser Schnittstelle lautet: <https://services.bgr.de/rest/services/geologie/gbl/MapServer>

Der MapService bildet die vollständige Schnittstelle ab: http://resources.arcgis.com/en/help/arcgis-rest-api/index.html#/Map_Service/02r3000000w2000000/

3.1.2 OGC WMS

Die Basis-URL dieser Schnittstelle lautet: <https://services.bgr.de/wms/geologie/gbl/>

Der MapService bildet die vollständige OGC Web Map Service Schnittstelle ab (<https://server.arcgis.com/de/server/10.3/publish-services/windows/ogc-support-in-arcgis-server.htm>).

3.2 Technische Umsetzung

3.2.1 Programmablauf

Diese Sicht zeigt, wie die verschiedenen Komponenten und beteiligten Nachbarsysteme in einem bestimmten Anwendungsfall miteinander interagieren. Um die Darstellung nicht zu überfrachten, wurden die Infrastrukturkomponenten in der Darstellung nicht berücksichtigt.

3.2.1.1 OGC WMS GetFeatureInfo (Einzellokation)

Der Abruf der OGC WMS GetFeatureInfo Operation soll nicht den Standard Response zurückliefern, sondern die Detailinformationen der Einzellokationen. Der Ablauf (technisch beschrieben im Sequenzdiagramm 'WMS GetFeatureInfo.png') ist wie folgt:

- Ein externer Viewer ruft das OGC WMS GetCapabilities des Dienstes ab. Der MapService liefert nicht das default Capabilities Dokument zurück, sondern ein übersteuertes Dokument, in welchem die URL auf die GetFeature Operation angepasst worden ist.
- Der externe Viewer würde nun bei jedem GetFeatureInfo Aufruf diese manipulierte URL verwenden. Die Manipulation besteht darin, dass dort ein weiterer Requestparameter angefügt ist, welcher auf ein XSL-Stylesheet verweist.

-
- Wenn der ArcGIS for Server eine solche GetFeatureInfo Anfrage verarbeiten muss, ruft der ArcGIS for Server in diesem Zuge das XSL Stylesheet von der URL ab. Anschließend transformiert der MapService den GetFeatureInfo Response mit diesem XSL Stylesheet
 - Das Stylesheet führt ein HTTP Redirect durch. Dieser Redirect führt zu dem Bohrpunkt Backend Profil.