# Stoercode.de API

Dokumentation zur Einbindung in 3rd-Party-Systeme  
Version 1.0 (22.06.2020)

Inhaltsverzeichnis

[Stoercode.de API 1](#_Toc43592945)

[Allgemeines 2](#_Toc43592946)

[Vertraulichkeit & Rechtliches 2](#_Toc43592947)

[API-Design 2](#_Toc43592948)

[API URL 2](#_Toc43592949)

[Zugriff auf API-Ressourcen & API-Authentifizierung 2](#_Toc43592950)

[Rückgabeformat 3](#_Toc43592951)

[Error Codes 3](#_Toc43592952)

[Fehlermeldungen 3](#_Toc43592953)

[API Funktionen 4](#_Toc43592954)

[getCategories – Alle Kategorien auslesen 4](#_Toc43592955)

[getErrorcodes – Fehlercodes zu einer Kategorie 6](#_Toc43592956)

[getSpareparts – Ersatzteilliste(n) zu einer Kategorie 7](#_Toc43592957)

[getPing – Testfunktion für Erreichbarkeit und Login 8](#_Toc43592958)

[Tipps & Tools 9](#_Toc43592959)

# Allgemeines

Mit der Stoercode.de-API können Sie die Inhalte von Stoercode.de in andere Anwendungen integrieren. Diese können Desktop- oder Webbasiert sein.

## Vertraulichkeit & Rechtliches

Die Inhalte dieses Dokuments sowie alle weiteren Informationen, die Ihnen in Bezug auf diese API, unseren Datenmodellen und Funktionen zugänglich und bekannt sind unterliegen strenger Vertraulichkeit. Eine Weitergabe an weitere Personen ist ohne ausdrückliche Genehmigung strikt untersagt.

Daten aus unseren Systemen sind unser Eigentum, eine Speicherung- und/oder Nutzungsoption wird ihnen gesondert zu bestimmten zwecken eingeräumt. Diese Berechtigung erhalten Sie schriftlich und ist binden. Eine Nutzung oder Speicherung darüber hinaus ist strikt untersagt.

Bei Einbindung in Drittsysteme ist sind immer die Anmeldedaten des Individuellen Kunden (dieser benötigt einen eigenen Storcode.de Account) zu übergeben. Eine Übergabe einer allgemeinen Kennung ist untersagt. Ebenfalls ist strikt drauf zu achten das Daten stets vor Zugriff dritter zu Schützen sind. Eine Verschlüsselte Ablage wird dringend empfohlen.

## API-Design

Die Stoercode.de-API ist eine HTTP-basierte API, was bedeutet, dass alle Interaktionen über Standard-HTTP stattfinden.

Alle Anforderungen an die API werden per POST Methode übermittelt.

Die API unterstützt den Export von Daten im JSON Format.

## API URL

Der Zugriff auf die API erfolgt über die URL:   
<https://www.stoercode.de/api-connect/3rdpartyconnector.php>

Bitte verwenden Sie unbedingt eine Verschlüsselte https Verbindung. Die Daten werden sonst unverschlüsselt und auslesbar über das Internet übermittelt.

## Zugriff auf API-Ressourcen & API-Authentifizierung

Ein Zugriff auf die API erfolgt durch Authentifizierung mit einer gültigen Benutzerkennung, welche bei jedem Aufruf mitgegeben werden muss.

Mindestens müssen immer per POST folgende Parameter mitgegeben werden:

|  |  |
| --- | --- |
| Parameter | Beschreibung |
| user | Benutzername (E-Mail) |
| pass | Passwort |
| action | Funktion der API die genutzt werden soll |

## Rückgabeformat

JSON  
Die Exporte enthalten in der Regel einen Block „result“ und einen „meta“. In „result“ finden sich die Daten üblicherweise im Listenformat. In „meta“ wiederrum allgemeine Informationen zu Fehlern, Anzahl der Zeiten oder auch eine Meldung in Worten.

Ein Beispiel für eine Kategorie:

{

"result": [

{

"id": "1",

"label": "Viessmann",

"parent\_id": "0",

"sort\_order": "99"

} ],

"meta": {

"error": "000",

"rows": 1,

"message": "OK"

}

}

## Error Codes

Neben dem beschreibenden Fehlertext enthalten Fehlernachrichten maschinenlesbare Codes. Während sich der Text für eine Fehlermeldung ändern kann, bleiben die Codes gleich. In der folgenden Tabelle werden die Codes beschrieben, die beim Arbeiten mit der API angezeigt werden können:

| **Code** | **Beschreibung** |
| --- | --- |
| **000** | OK – Kein Fehler |
| **001** | Es wurden keine Zeilen zur Abfrage gefunden |
| **…** | … |
| **998** | Es wurde keine „action“ (Function) angegeben. |
| **999** | Keine Berechtigung – User inaktiv oder falsch |

## Fehlermeldungen

Wenn die Stoercode.de API Fehlermeldungen zurückgibt, geschieht dies im „meta“ Bereich der JSON Antwort. Ein Fehler könnte beispielsweise so aussehen:

{"meta": {

"error": "999",

"rows": 0,

"message": "you are not allowed"

}}

# API Funktionen

## getCategories – Alle Kategorien auslesen

**POST: ?user=<user>&pass=<pass>&action=getCategories**(ACHTUNG: Dieser Aufruf dient nur zur Veranschaulichung welche Parameter an die API per POST übergeben werden müssen. Ein Zugriff per GET ist nicht vorgesehen.)

Mit dieser Funktion erhalten Sie alle unsere Kategorien, welche zu einem Geräte-Baum logisch zusammengefügt werden können. Die Verknüpfung erfolgt immer per partent\_id in welcher die jeweils höherliegende Kategorie-ID steht.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | label | parent\_id |  | Erklärung |
| 100 | Hersteller A | 0 |  | 0 = Oberste Ebene |
| 110 | Ölbrenner | 100 |  | 100 = Kategorie gehört unter Hersteller A (100) |
| 212 | Gerät Öl-A | 110 |  | 110 = Kategorie gehört zu Ölbrenner (110) |
| 112 | Gaskessel | 100 |  | 100 = Kategorie gehört unter Hersteller A (100) |
| 113 | Gerät Gas-A | 112 |  | 112 = Kategorie gehört zu Gaskessel (112) |
| 114 | Gerät Gas-B | 112 |  | 112 = Kategorie gehört zu Gaskessel (112) |
| 232 | Hersteller B | 0 |  | 0 = Oberste Ebene |

|  |  |
| --- | --- |
| id | ID (eindeutig) der Kategorie |
| label | Bezeichnungstext |
| parent\_id | Übergeordnete Kategorie-ID, Eltern-ID der jeweiligen Kategorie.  Bei “0” handelt es sich um die oberste Ebene. |
| sort\_order | Sortierungskennzahl |
| error | Fehlermeldung, wenn > 0 ansonsten OK; Beschreibung siehe weiter oben bei Error Codes |
| rows | Anzahl der Zeilen in “result” |
| message | Fehlermeldung in Textform |

JSON Response:

{

"result": [

{

"id": "1",

"label": "Viessmann",

"parent\_id": "0",

"sort\_order": "99"

}, << weitere … >>

],

"meta": {

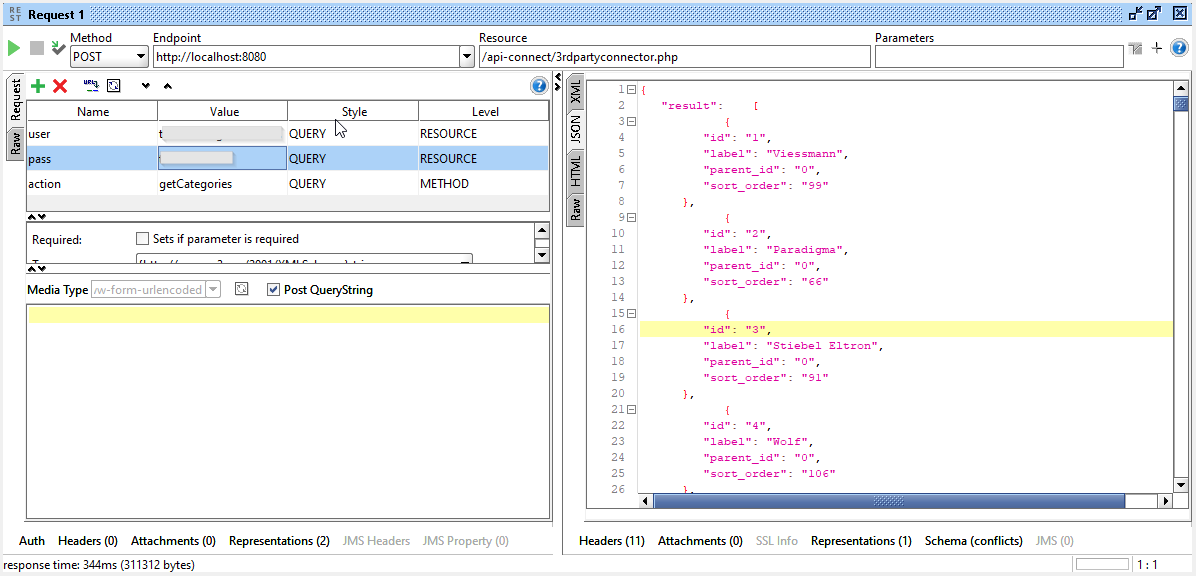
"error": "000",

"rows": 4000,

"message": "OK"

}

}



## getErrorcodes – Fehlercodes zu einer Kategorie

**POST: ?user=<user>&pass=<pass>&action=getErrorcodes&cat=<catID>**(ACHTUNG: Dieser Aufruf dient nur zur Veranschaulichung welche Parameter an die API per POST übergeben werden müssen. Ein Zugriff per GET ist nicht vorgesehen.)

Mit dieser Funktion können Sie alle Fehlercodes unserer Datenbank abrufen, die genau einer Kategorie zugeordnet sind. Sinnvollerweise sollte hier eine ID auf Geräteebene angegeben werden. Die ID erhalten Sie aus einem vorherigen Abruf unserer Kategorien.

|  |  |
| --- | --- |
| id | ID (eindeutig) des Fehlercodes |
| label | Bezeichnungstext |
| content | Fehlerbeschreibung im HTML Format. Es können Bilder im Base64 Format enthalten sein. |
| sort\_order | Sortierungskennzahl |
| error | Fehlermeldung, wenn > 0 ansonsten OK; Beschreibung siehe weiter oben bei Error Codes |
| rows | Anzahl der Zeilen in “result” |
| message | Fehlermeldung in Textform |

JSON Response:

{

"result": [

{

"id": "65662",

"label": "80 00",

"content": "<b>Fehler: 80 00…<br> ",

"sort\_order": "1"

}, << weitere … >>

],

"meta": {

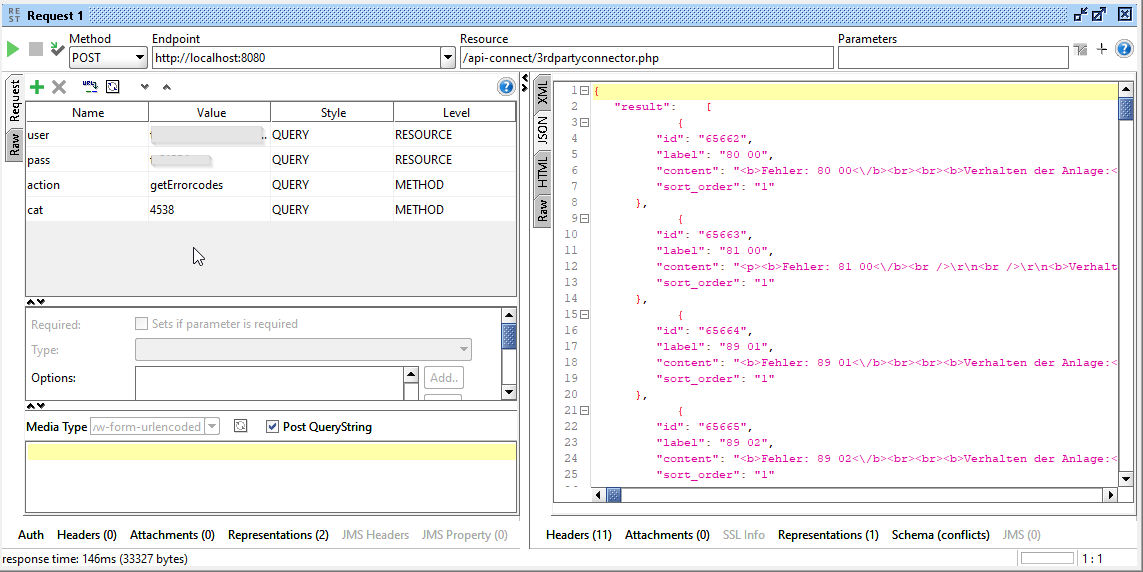
"error": "000",

"rows": 67,

"message": "OK"

}

}



## getSpareparts – Ersatzteilliste(n) zu einer Kategorie

**POST: ?user=<user>&pass=<pass>&action=getSpareparts&cat=<catID>**(ACHTUNG: Dieser Aufruf dient nur zur Veranschaulichung welche Parameter an die API per POST übergeben werden müssen. Ein Zugriff per GET ist nicht vorgesehen.)

Mit dieser Funktion können Sie alle Ersatzteillisten abrufen, die genau einer Kategorie zugeordnet sind. Sinnvollerweise sollte hier eine ID auf Geräteebene angegeben werden. Die ID erhalten Sie aus einem vorherigen Abruf unserer Kategorien. In der Regel wird hier nur ein Datensatz, in dem die Listen im content per html verlinkt sind ausgegeben. Es kann aber auch keine Listen geben.

|  |  |
| --- | --- |
| id | ID (eindeutig) des Fehlercodes |
| label | Bezeichnungstext |
| content | Fehlerbeschreibung im HTML Format. Es können Bilder im Base64 Format enthalten sein. |
| sort\_order | Sortierungskennzahl |
| error | Fehlermeldung, wenn > 0 ansonsten OK; Beschreibung siehe weiter oben bei Error Codes |
| rows | Anzahl der Zeilen in “result” |
| message | Fehlermeldung in Textform |

JSON Response:

{

"result": [ {

"id": "19562",

"label": "Ersatzteilliste",

"content": "<p>Direktlink zur Ersatzteilliste…",

"sort\_order": "1"

}],

"meta": {

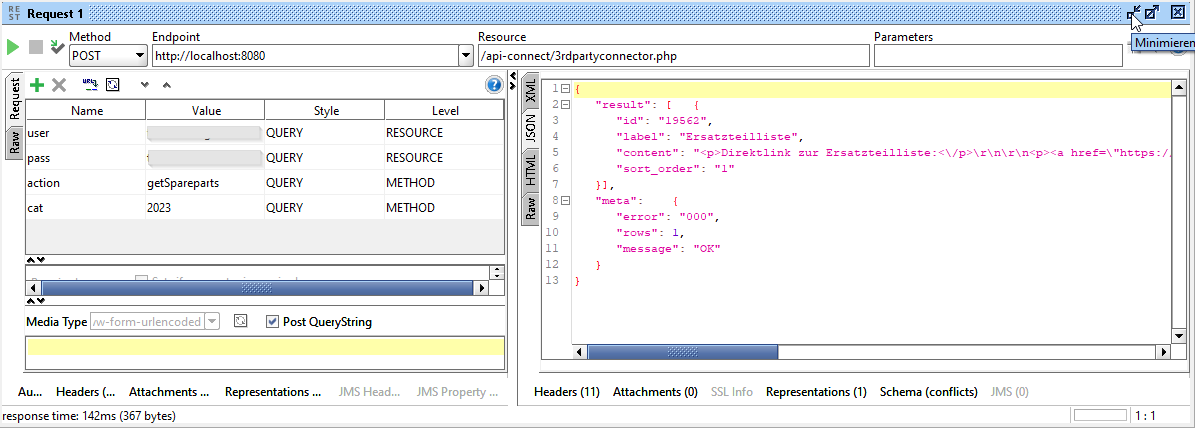
"error": "000",

"rows": 1,

"message": "OK"

}

}



## getPing – Testfunktion für Erreichbarkeit und Login

**POST: ?user=<user>&pass=<pass>&action=getPing**(ACHTUNG: Dieser Aufruf dient nur zur Veranschaulichung welche Parameter an die API per POST übergeben werden müssen. Ein Zugriff per GET ist nicht vorgesehen.)

Diese Funktion kann Ihnen zu Testzwecken oder auch im produktiven Betrieb hilfreich sein. Es ist damit möglich einen Login zu prüfen oder auch die Erreichbarkeit des Systems.

|  |  |
| --- | --- |
| id | ID (eindeutig) des Fehlercodes |
| label | Bezeichnungstext |
| content | Fehlerbeschreibung im HTML Format. Es können Bilder im Base64 Format enthalten sein. |
| sort\_order | Sortierungskennzahl |
| error | Festwert 000 wenn aufgerufen; Enthält keine Funktionen die Fehler hervorrufen können. |
| rows | Festwert 1 |
| message | Festwert 1 |

JSON Response:

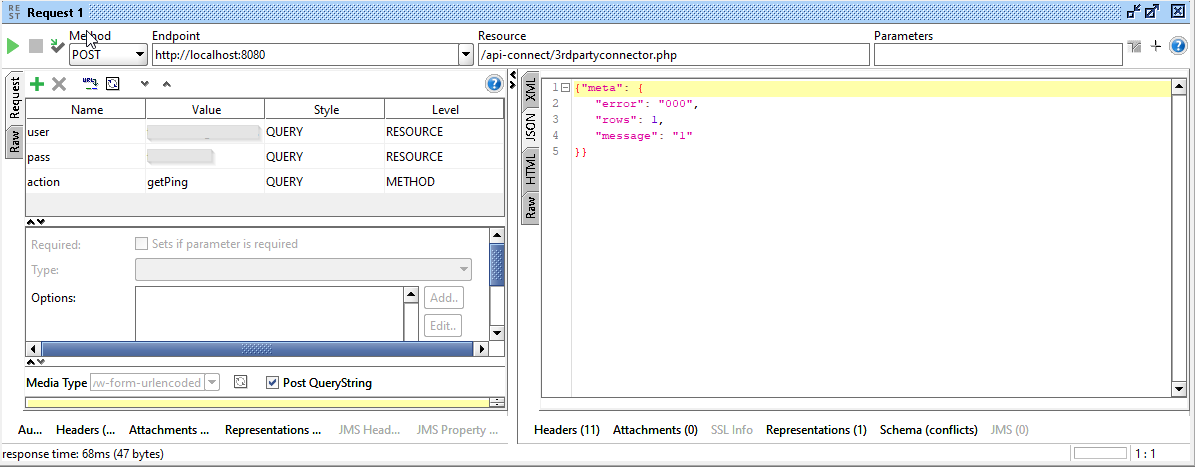
{"meta": {

"error": "000",

"rows": 1,

"message": "1"

}}



# Tipps & Tools

Zum Testen unserer API kann z.B. SoapUi verwendet werden. Dies kann bei der Fehlersuche hilfreich sein.

Aufgrund der Vielzahl von möglichen Zielsystemen sehen wir von der allgemeinen Bereitstellung von Demoprojekten ab. Die Verwendung von HTTP