

# **epoq Recommendation Service**

## ***Schnittstellendokumentation***

Version 3.1  
© epoq GmbH  
Mai 2012

## Inhaltsverzeichnis

1. Änderungsindex .....	3
2. Funktionserklärung.....	4
2.1. Preface.....	4
2.1.1. Sammeln der Daten .....	4
2.1.2. Katalog.....	4
2.1.3. Anzeige der Empfehlungen.....	4
2.2. Feinheiten .....	5
2.2.1. Varianten.....	5
2.2.2. Session Id's.....	5
2.2.3. Caching.....	6
2.2.4. SEO / duplicate Content.....	6
2.2.5. Timeout Strategie.....	6
2.2.6. Filter / Regeln.....	6
3. Technische Integration .....	7
3.1. Aufrufparameter .....	7
3.1.1. Pflichtparameter.....	7
3.1.2. Zusätzliche Parameter für die Übergabe der Artikelstruktur an epoq.....	8
3.1.3. Parameter für Abholen der Empfehlungen.....	8
3.2. Einbau des Klick und Kauf Tracking .....	9
3.2.1. viewItem – Artikel betrachten auf der Produktdetail-Seite.....	9
3.2.1.1. REST-API.....	9
3.2.1.2. Web-API.....	9
3.2.2. updateCart - Warenkorb aktualisieren .....	9
3.2.2.1. REST-API.....	9
3.2.2.2. Web-API.....	9
3.2.3. ProcessCart – Warenkorb kaufen .....	10
3.2.3.1. REST-API.....	10
3.2.3.2. Web-API.....	10
3.3. Anzeige der Empfehlungen.....	12
3.3.1. Format der Empfehlungen .....	12
3.3.1.1. schlankes XML (nur Produkt-Id's).....	12
3.3.1.2. erweitertes Format mit zusätzlichen Attributen aus dem Katalog (JSON oder XML).....	14
3.3.2. Klickfeedback der Empfehlungen.....	15
3.3.3. Produktdetailseite .....	15
3.3.3.1. REST-API.....	15
3.3.3.2. Web-API.....	16
3.3.3.3. AJAX/JSON.....	16
3.3.4. Warenkorb .....	19
3.3.4.1. REST-API.....	19
3.3.4.2. Web-API.....	19
3.3.4.3. AJAX/JSON.....	19
3.3.5. Startseite.....	20
3.3.5.1. REST-API.....	20
3.3.5.2. Web-API.....	20
3.3.5.3. AJAX/JSON.....	20
4. Best Practices / FAQ .....	21
4.1. Empfehlungen im Newsletter oder basierend auf Kundendaten .....	21
4.2. Einbau der Empfehlungen .....	21
4.3. Sonstiges .....	21

## 1. Änderungsindex

Datum	Autor	Beschreibung der Änderung	Version
10.05.2010	S. Kölle		2.1.1
07.09.2010	D. Schalk	Fehlerkorrektur, neue Screenshots, Entkopplung Katalogformate, Entkopplung REST-API	2.2
31.05.12	D. Schalk	Ergänzung neuer Funktionen	3.0
04.06.12	D. Schalk	Erweiterung AJAX Dokumentation	3.1

## 2. Funktionserklärung

### 2.1. Preface

Der Recommendation Service (epoq RS) generiert auf Basis des Klick- und Kaufverhaltens aller Besucher und Kunden passende Produktempfehlungen. Diese können auf unterschiedlichen Seiten des Shops angezeigt werden, um dem Webshop Besucher beratend zur Seite zu stehen. Die Empfehlungen auf Produktdetailseiten werden auf dem Klick- und Kaufverhalten der kompletten Webshop Besucher generiert, wobei auf der Startseite eher auf das persönliche Verhalten eines einzelnen Besuchers geachtet wird.

Die Einbindung von epoq RS erfolgt in drei Phasen :

- Einbau des Trackings (Übertragung der Klicks und Käufe an epoq)
- Aufbau der Datenbasis bei epoq (Lernphase)
- Einbau der Empfehlungen im Shop

#### 2.1.1. Sammeln der Daten

Da epoq RS Empfehlungen anhand des Klick- und Kaufverhaltens der Webshop Besucher generiert, benötigt der Service die kompletten Session-Daten der Besucher. Dafür stellt epoq RS Standardfunktionen bereit, die in den Shop integriert werden müssen.

Bevor das Tracking eingebaut wird, und somit das Besucherverhalten an epoq übertragen wird, muss klar sein, was Sie Ihren Besuchern im Shop anbieten möchten. Möchten Sie individuelle Empfehlungen (z.B. auf der Startseite oder im Newsletter), so müssen ein paar Dinge von Anfang an berücksichtigt werden. Näheres wird in den nächsten Abschnitten erklärt.

Die erste Phase der Einbindung von epoq RS betrifft die Übertragung des Kundenverhaltens im Shop. Sobald die Daten eine gute Signifikanz erreichen, kann die Anzeige der Empfehlungen eingebaut werden.

#### 2.1.2. Katalog

Für eine Gewährleistung höchster Datenqualität und Datenaktualität ist die Bereitstellung eines Produktkatalogs empfehlenswert. Dieser sollte mindestens einmal am Tag generiert werden. Je nach Kataloginhalt und Wichtigkeit der Aktualität kann es notwendig sein, einen Katalog auch mehrmals täglich zur Verfügung zu stellen.

Bei Magento und Oxid-Shop Nutzern, die unsere Plugins verwenden, möchten wir darauf hinweisen, dass der Export schrittweise erfolgt, da der Katalog in einzelnen Teilen generiert wird, wodurch sich die Dauer des Imports erhöht. Hier besteht auch die Möglichkeit, nicht den Plugin-Export, sondern einen anderen Katalog, wie z.B. den Katalog aus dem führenden System (Warenwirtschaft o.ä.) zu nutzen.

Im Allgemeinen sollte die Katalogdatei auf einem http- oder ftp-Server zur Verfügung gestellt werden und kann, bei Bedarf, mit einer Standardauthentifizierung gesichert sein.

Weitere Informationen zu den Formaten und Feldern gibt es in dem Dokument `eServices_Katalogformat_xxx_deutsch.pdf`

Bei epoq RS können manche Filterfunktionen, zum Beispiel nach „in der Kategorie“ oder „Zubehör“, nur fehlerfrei genutzt werden, wenn alle Felder des Katalogs entsprechend ausgefüllt sind.

#### 2.1.3. Anzeige der Empfehlungen

Sie können die Empfehlungen in verschiedene Seiten Ihres Webshops einbinden. Die Empfehlungen sollten allerdings erst angezeigt werden, wenn genügend Daten bei epoq gesammelt wurden und die Signifikanz der Empfehlungen hoch genug ist. Der Zeitpunkt zum Livegang wird Ihnen von epoq mitgeteilt.

Vor dem Livegang können die Empfehlungen im epoq Kundenlogin geprüft werden. Gerne zeigen wir Ihnen die verschiedenen Filtermöglichkeiten, die es gibt, um die Empfehlungen bei Bedarf nach Ihren Vorstellungen anzupassen.

Standardmäßig werden immer 6 Empfehlungen geliefert. Dies können wir gerne anpassen. Teilen Sie uns einfach mit wie viele Empfehlungen Sie in Ihrem Shop anzeigen möchten.

epoq RS bietet auch die Möglichkeit, mehrere Empfehlungsblöcke zu beliefern. Beachten Sie hierzu auch folgenden Punkt 3.3.

## 2.2. Feinheiten

### 2.2.1. Varianten

Falls Sie in Ihrem Produktsortiment Hauptprodukte und Varianten pflegen, so müssen bei der Integration ein paar Dinge beachtet werden. Generell sollten Sie uns die Information übermitteln, ob es sich bei einem angeschauten, angeklickten oder gekauften Produkt um eine Variante oder ein Hauptprodukt handelt. Diese Informationen sollten nicht nur beim Tracking berücksichtigt werden, sondern auch im Katalog zur Verfügung stehen.

Wir empfehlen hierbei folgendes :

- Falls es Größen und Farbvarianten eines Produktes gibt, so sollten :
  - jede Farbausführung als eigenes Produkt im Katalog-Export vorhanden sein. Dieses Produkt sollte eine Referenz auf das Hauptprodukt enthalten (*variantOf=<HauptproduktId>*) und die verfügbaren Größen ausweisen: „*Größe:Verfügbare Anzahl:Preis#Größe:Verfügbare Anzahl:Preis#Größe:Verfügbare Anzahl:Preis*“. Wenn eine Größe zwar verfügbar war, aber der Bestand auf 0 ist, soll dieses auch so exportiert werden.
  - beim Anschauen eines Artikels, die jeweilige Hauptprodukt-Id übermitteln werden (*productId=<HauptproduktId>*).
  - beim Ändern des Warenkorb und Kauf eines Artikels die Hauptprodukt-Id und auch die Artikel-Id übergeben werden (*productId=<ArtikelId>&variantOf=<HauptproduktId>*)
  - entweder Hauptprodukte oder Farbvarianten als Empfehlungen angezeigt werden.
- Falls ein Hauptprodukt alle Größen und Farben enthält (nur eine Produkt-Id), so ist es epoq leider nicht möglich die Empfehlungen auf Größen und/oder Farben einzuschränken, da nicht unterschieden werden kann welche Größe oder Farbe sich ein Besucher anschaut.

### 2.2.2. Session Id's

Es ist sehr wichtig und zwingend erforderlich, dass jeder Webshop Besucher eine eindeutige Session Id bekommt. Die SessionId dient dazu, das Klick- und Kaufverhalten eines Besuchers zu identifizieren, um anhand dieser zusammenhängenden Informationen qualitativ gute Empfehlungen zu generieren.

Es ist oftmals der Fall, dass ein Besucher erst dann eine SessionId bekommt, wenn er ein Produkt in den Warenkorb legt. Dies ist für epoq RS aber schon zu spät, da sein vorheriges Klickverhalten nicht identifiziert werden kann und somit verloren geht. In diesem Fall kann nur auf das Kaufverhalten gelernt werden, was bedeutet, dass der größte Teil an Informationen verloren geht. Die Einstellung, wie und wann eine SessionId generiert wird, kann normalerweise in Ihrem Shopsystem vorgenommen werden. Wenden Sie sich hier am Besten an Ihre Agentur oder den Shophersteller.

Oftmals werden Session Id's nicht eindeutig pro Besucher vergeben, weil ein Caching-System im Einsatz ist. Bitte lesen Sie hierzu den nächsten Abschnitt durch.

### 2.2.3. Caching

Generell passen Individualisierung (persönliche Empfehlungen) und Caching nicht zusammen, weil der Zweck beider Dinge ganz verschieden ist. Gecachte Empfehlungen sind in der Regel nicht mehr individuell. Der Einsatz eines Caching Systems lässt es meistens nicht zu, eine eindeutige und durchführende Session Id pro Webshop Besucher zu vergeben.

Falls Sie ein Caching System im Einsatz haben, sollten Sie sich für die AJAX/JSON Variante der Integration entscheiden. Diese wird in Punkt 3.3.3.3 dargestellt.

### 2.2.4. SEO / duplicate Content

Die Integration von epoq RS kann auch Auswirkungen auf SEO (*Search Engine Optimization* : Suchmaschinenoptimierung) haben. Um ungewollte Nebeneffekte, wie z.B. „duplicate Content“ (der Inhalt der Seite kann über mehrere URL's erreichbar sein) zu vermeiden, empfehlen wir Ihnen im Kopf Ihrer Seiten den Parameter

```
< link rel="canonical" href="http://www.example.com/product.php?item=swedish-fish" / >
```

zu setzen, welcher die maßgebliche URL des Produkts festlegt und von Google & Co ausgelesen werden kann.

### 2.2.5. Timeout Strategie

Da es bei unserem Service, wie auch bei allen anderen gehosteten Services, allein schon durch die Verbindung über das öffentliche Netz, hin und wieder zu Verzögerungen in der Antwortzeit kommen kann, empfehlen wir dringend Vorkehrungen zu treffen, um kein zusätzliches Ausfallrisiko für den Webshop darzustellen.

Daher sollten grundsätzlich Timeouts für alle Aufrufe des epoq Recommendation Service festgelegt werden. Diese stellen sicher, dass bei Überschreiten einer bestimmten Antwortzeit nicht länger auf die Antwort des Services gewartet wird und die Seite ohne die vom Service erwarteten Informationen aufgebaut wird.

Die Länge der Timeouts sollte je nach Anforderung zwischen 500 Millisekunden und 2 Sekunden liegen. Im Normalfall beantwortet unser Service alle Requests in weniger als 100 Millisekunden.

### 2.2.6. Filter / Regeln

epoq RS liefert zahlreiche Empfehlungen, die das persönliche Kaufverhalten der Kunden im Shop abbilden. In vielen Fällen ist es sinnvoll, die Vielfalt der Empfehlungen auf die Empfehlungsstelle im Shop anzupassen. Hierzu kann epoq Regeln für Sie hinterlegen, die die Empfehlungen filtern oder in manchen Fällen auch erweitern.

Regeln können sowohl Empfehlungen filtern, als auch das Verhalten des Recommendation Service ändern.

Diese sind an zwei Stellen konfigurierbar:

- Als zusätzlicher oder veränderter Parameter bei der Anfrage nach einer Empfehlung. Hierzu muss der Parameter „&rules=REGEL“ übergeben werden( z.B rules=InCategoryFilter)
- Als feste Konfiguration für Ihren Shop. Der epoq Support richtet dies auf Anfrage für Sie ein.

Soweit es sich bei den Regeln um Filter handelt, können diese beliebig kombiniert werden. Jeder Filter schränkt Empfehlungen weiter ein. Wenn es sich bei einer Regel um eine Erweiterung handelt (z.B. hinzufügen von manuellen Empfehlungen) kann es gewisse Einschränkungen geben, falls Sie dazu einen Filter nutzen wollen. Bitte kommen Sie auf uns zu und teilen Sie uns Ihre Vorstellungen mit.

epoq bietet im Kundenlogin die Möglichkeit den Einfluss von Regeln vor der Anwendung im Live-Betrieb zu erproben und damit sicher zu stellen, dass die Wahrnehmung des Kunden auch Ihrem Wunsch entspricht.

### 3. Technische Integration

Für jede epoq RS Funktion gibt es verschiedene Möglichkeiten, diese in den Shop zu integrieren. Wir werden auf jede Möglichkeit in den einzelnen Punkten eingehen. Hier ein Überblick:

- Einbindung über **REST-API** (direkte http-Calls aus dem Shop-Backend)
- Integration über **Web-API** (JavaScript und HTML): Die Service-Aufrufe werden direkt im HTML-Code des Webshops eingebunden. Zunächst muss in die Webshop Templates im <head> Element das epoq-Skript eingebunden werden:

```
<script type="text/javascript" src="http://rs.epoq.de/web-api/epoq.js"></script>
```

oder bei verschlüsselter Kommunikation:

```
<script type="text/javascript" src="https://rs.epoq.de/web-api/epoq.js"></script>
```

Sie können das Skript auch direkt in das <body> Element, bei dem jeweiligen Aufruf der Funktion einfügen.

- Einbau von **AJAX/JSON** Calls (diese sind nur für das Abholen der Empfehlungen relevant)

Um die Empfehlungen per AJAX Call anzuzeigen benötigen Sie das JavaScript epoqgo.js. Dieses müssen Sie wie folgt in den <head>-Tag der Seite einbinden :

```
<script type="text/javascript" src="http://rs.epoq.de/web-api/epoqgo.js"></script>
```

Falls die Kommunikation mit epoq verschlüsselt werden soll, muss das Skript wie folgt eingebunden werden :

```
<script type="text/javascript" src="https://rs.epoq.de/web-api/epoqgo.js"></script>
```

#### 3.1. Aufrufparameter

##### 3.1.1. Pflichtparameter

Bei den hier genannten Aufrufparametern kann es sich um beliebige Zeichenketten (Strings) handeln:

REST-API / JSON	Web-API	Beschreibung
tenantId	epoq_tenantId	Die eindeutige Identifikation des Webshops (wird Ihnen von epoq zugeteilt)
customerId	epoq_customerId	Eine eindeutige Kundennummer, kann einigen Funktionen übergeben werden sobald diese bekannt ist - der Kunde sich also dem Shop gegenüber authentifiziert hat (z.B. per Login bei Bestellung, Neuregistrierung, Cookie),- sollte die customerId bei allen Aufrufen an epoq RS übergeben werden. Wenn die Kundennummer nicht bekannt ist, so kann der Parameter leer übergeben oder komplett weggelassen werden.
sessionId	epoq_sessionId	Eine eindeutige Kennzeichnung der Session eines Kunden im Webshop

Die Parameter customerId und sessionId können auf der Seite des Webshops in vielen verschiedenen Formaten vorliegen und müssen eventuell für die epoq RS-Aufrufe in eine eindeutige String-Repräsentation umgewandelt werden.

### 3.1.2. Zusätzliche Parameter für die Übergabe der Artikelstruktur an epoq

REST-API / JSON	Web-API	Beschreibung
productId	epoq_productId	Eine eindeutige Kennzeichnung des Produktes
quantity	epoq_quantity	Die Anzahl des Produktes. Wird das Feld leer gelassen, wird implizit 1 angenommen.
price	epoq_price	Der Produkt Preis. Generell sind zwei Nachkommastellen erlaubt. Dezimaltrenner ist '.' Beispiel: epoq_price = '4.60';
variantOf	epoq_variantOfList	Produkt Id des Hauptproduktes, falls es Varianten gibt
	epoq_tag	Der Parameter muss bei Nutzen eines Cookies mit dem Wert „true“ übergeben werden.

### 3.1.3. Parameter für Abholen der Empfehlungen

Erforderlich	REST-API /JSON	Web-API	Beschreibung
x	productId	epoq_productId	Eine eindeutige Kennzeichnung des Produktes
x	tenantId	epoq_tenantId	Die eindeutige Identifikation des Webshops (wird Ihnen von epoq zugeteilt)
x	sessionId	epoq_sessionId	Eine eindeutige Kennzeichnung der Session eines Kunden im Webshop
optional <b>Nicht im Produktiv-Betrieb verwenden!</b>	demo	epoq_demo	Zum Testen des Layouts können hiermit zufällige Empfehlungen angefordert werden. Dies funktioniert erst wenn der Katalog geladen ist.
x	widgetTheme	epoq_widgetTheme	Es existieren verschiedene Widget-Typen. (Erklärung siehe 3.3.1)
optional	locakey	epoq_locakey	Wenn es sich um einen mehrsprachigen Webshop handelt, muss hier ein 'key' für die Sprache übergeben werden, in der die Seite gerade angezeigt wird. Beispiel: epoq_locakey = 'en';
x	rules	epoq_rules	Hier können Regeln übergeben werden wie z.B. InCategoryFilter  Wenden Sie sich hier bitte an den epoq Support wenn Sie spezielle Regeln einstellen möchten



## 3.2. Einbau des Klick und Kauf Tracking

### 3.2.1. viewItem – Artikel betrachten auf der Produktdetail-Seite

Hier geht es darum epoq mitzuteilen, dass Artikel angeschaut wird. Hierfür gibt es zwei Möglichkeiten. Die Integration kann über REST oder Web-API erfolgen.

Folgende Parameter sind gültig (siehe 3.1): `productId`, `tenantId`, `sessionId`, `variantOf`.

#### 3.2.1.1. REST-API

In das Template sollte dieser Aufruf eingebaut werden. Die Werte müssen entsprechend ersetzt werden :

```
http://rs1.epoq.de/inbound-servletapi/viewItem?tenantId=IHRE_EPOQ_KENNUNG
&sessionId=SESSION_ID&productId=PRODUKT_ID&variantOf=HAUPTPRODUKT_ID
```

#### 3.2.1.2. Web-API

Das Webshop Template, das zum Anzeigen eines Produktes verwendet wird (Produktdetail-Ansicht) muss einen Eintrag nach folgendem Beispiel erhalten:

```
<script type="text/javascript">
    epoq_tenantId      = 'IHRE_EPOQ_KENNUNG';
    epoq_sessionId     = 'SESSION_ID';
    epoq_productId     = 'PRODUKT_ID';
    epoq_variantOfList = 'HAUPTPRODUKT_ID';
    epoq_viewItem();
</script>
```

### 3.2.2. updateCart - Warenkorb aktualisieren

Wenn sich der Warenkorb eines Kunden ändert (Neuer Artikel hinzufügen, Anzahl ändern, Artikel löschen), muss die Funktion aufgerufen werden. Nach jeder Änderung muss der gesamte Warenkorb mit allen Artikeln übertragen werden.

Folgende Parameter sind gültig (siehe 3.1): `productId`, `tenantId`, `sessionId`, `variantOf`, `quantity`, `price`.

#### 3.2.2.1. REST-API

In das Template sollte dieser Aufruf eingebaut werden. Die Werte müssen entsprechend ersetzt werden :

```
http://rs1.epoq.de/inbound-servletapi/updateCart?tenantId=IHRE_EPOQ_KENNUNG
&customerId=KUNDEN_KENNUNG&sessionId=SESSION_ID&productId=PRODUKT_ID1
&productId=PRODUKT_ID2&productId=PRODUKT_ID3&quantity=MENGE_PRODUKT1&quantity=MENGE_PRODUKT2&
quantity=MENGE_PRODUKT3&unitPrice=PREIS_PRODUKT1&unitPrice=PREIS_PRODUKT2&unitPrice=PREIS_PRO
DUKT3&variantOf=HAUPTPRODUKT_ID_VON_PRODUKT1&variantOf=HAUPTPRODUKT_ID_VON_PRODUKT2&variantOf=HA
UPTPRODUKT_ID_VON_PRODUKT3
```

#### 3.2.2.2. Web-API

Folgender JavaScript Code muss im Template eingebettet sein :

```
<script type="text/javascript">
    epoq_tenantId      = 'IHRE EPOQ KENNUNG';
    epoq_sessionId     = 'SESSION_ID';
    epoq_customerId    = 'CUSTOMER_ID';
```

```

epoq_productIds[0] = 'PRODUKT_ID1';
epoq_quantities[0] = 'MENGE_PRODUKT1';
epoq_unitPrices[0] = 'PREIS_PRODUKT1';
epoq_variantOfList[0] = 'HAUTPRODUKT_ID_VON_PRODUKT1';

epoq_productIds[1] = 'PRODUKT_ID2';
epoq_quantities[1] = 'MENGE_PRODUKT2';
epoq_unitPrices[1] = 'PREIS_PRODUKT2';
epoq_variantOfList[1] = 'HAUTPRODUKT_ID_VON_PRODUKT2';

epoq_productIds[2] = 'PRODUKT_ID3';
epoq_quantities[2] = 'MENGE_PRODUKT3';
epoq_unitPrices[2] = 'PREIS_PRODUKT3';
epoq_variantOfList[2] = 'HAUTPRODUKT_ID_VON_PRODUKT3';
...
epoq_updateCart();
</script>

```

### 3.2.3. ProcessCart – Warenkorb kaufen

Die Käufe sind ein sehr wichtiger Bestandteil für den Recommendation Service, da ein Kauf eine eindeutige Zusage des Kunden ist. Hier sollte bei dem processCart() Aufruf die Kundennummer der Käufers (falls bekannt) an epoq übertragen werden. Dies ist auch für die Nutzung der Empfehlungen im Newsletter erforderlich.

Die entsprechenden Aufrufe werden meistens in die « Danke-Seite » implementiert (nach dem Kauf).

Folgende Parameter sind gültig (siehe 3.1): productId, tenantId, sessionId, variantOf, quantity, price.

#### 3.2.3.1. REST-API

In das Template sollte dieser Aufruf eingebaut werden. Die Werte müssen entsprechend ersetzt werden :

```

http://rs1.epoq.de/inbound-servletapi/processCart?tenantId=IHRE_EPOQ_KENNUNG
&customerId=KUNDEN_KENNUNG&sessionId=SESSION_ID&productId=PRODUKT_ID1
&productId=PRODUKT_ID2&productId=PRODUKT_ID3&quantity=MENGE_PRODUKT1&quantity=MENGE_PRODUKT2&
quantity=MENGE_PRODUKT3&unitPrice=PREIS_PRODUKT1&unitPrice=PREIS_PRODUKT2&unitPrice=PREIS_PRO
DUKT3&variantOf=HAUTPRODUKT_ID_VON_PRODUKT1&variantOf=HAUTPRODUKT_ID_VON_PRODUKT2&variantOf=H
AUTPRODUKT_ID_VON_PRODUKT3

```

#### 3.2.3.2. Web-API

Im dem Template, das nach einer erfolgreichen Transaktion anzeigt wird (der Warenkorb wurde gekauft), muss JavaScript Code nach folgendem Beispiel eingebettet sein:

```

<script type="text/javascript">
    epoq_tenantId      = 'IHRE EPOQ KENNUNG';
    epoq_sessionId     = 'SESSION_ID';
    epoq_customerId    = 'CUSTOMER_ID';

    epoq_productIds[0] = 'PRODUKT_ID1';
    epoq_quantities[0] = 'MENGE_PRODUKT1';
    epoq_unitPrices[0] = 'PREIS_PRODUKT1';
    epoq_variantOfList[0] = 'HAUTPRODUKT_ID_VON_PRODUKT1';

```

```
epoq_productIds[1] = 'PRODUKT_ID2';
epoq_quantities[1] = 'MENGE_PRODUKT2';
epoq_unitPrices[1] = 'PREIS_PRODUKT2';
epoq_variantOfList[1] = 'HAUTPRODUKT_ID_VON_PRODUKT2';

epoq_productIds[2] = 'PRODUKT_ID3';
epoq_quantities[2] = 'MENGE_PRODUKT3';
epoq_unitPrices[2] = 'PREIS_PRODUKT3';
epoq_variantOfList[2] = 'HAUTPRODUKT_ID_VON_PRODUKT3';

epoq_processCart();
</script>
```

### 3.3. Anzeige der Empfehlungen

Die Empfehlungen werden an der Stelle in der Webseite angezeigt, an der die Abfrage eingebettet ist.

Es können Empfehlungen abgeholt werden für:

- ein einzelnes Produkt (`getRecommendations()?rules=default`). Diese können z. B. unter dem Produkt auf dessen Produkt-Detail-Seite angezeigt werden.
- den aktuellen Warenkorb des Besuchers (`getRecommendations()?rules=RecommendationForCartRule`). Diese können dem Besucher z.B. unter seinem Warenkorb angezeigt werden.
- einen Kunden basierend auf dessen gesamten Kundenhistorie (`getRecommendations()?rules=RecommendationForCustomerRule`). Diese können z. B. auf der Begrüßungsseite des Kunden oder auf der Seite, die der Kunden nach dem Login angezeigt bekommt, eingeblendet werden.
- einen Kunden basierend auf dessen Klick- und Kaufhistorie aus dem gespeicherten Cookie. Diese können z. B. auf der Startseite des Shops angezeigt werden. Dies kann nicht aus dem Backend erfolgen und sollte über AJAX/JSON implementiert werden.
- Mehrere Widgets gleichzeitig (`getRecommendations()?rules=RULE1;RULE2`). Hier ist es äußerst wichtig, dass nur ein `getRecommendations()` Aufruf bei der Anzeige von mehreren Widgets auf der gleichen Seite gesendet wird. Ein häufig vorkommender Fehler ist es z.B. zwei `getRecommendations()` Aufrufe für eine gleiche Seite zu schicken. Die Problematik besteht darin, dass epoq nur dann sicherstellen kann, dass beide Widgets Dubletten-frei sind, wenn nur ein Request geschickt wird.

Die verschiedenen Parameter zum Abholen der Empfehlungen sind unter Punkt 3.1.3 beschrieben.

#### 3.3.1. Format der Empfehlungen

Der Parameter `widgetTheme` definiert in welchem Format die Empfehlungen zurückgeliefert werden sollen. Hier ein kurzer Überblick der verschiedenen Möglichkeiten :

##### 3.3.1.1. schlankes XML (nur Produkt-Id's)

- Wenn der Parameter `widgetTheme=xml` übergeben wird, werden die Empfehlungen in einem einfachen XML Format geliefert. In der Datei sind die Produkt-Id's sowie eine `recommendationId` vorhanden.

Beispiel :

```
<?xml version='1.0' encoding='ISO-8859-1' standalone='yes'?>

<recommendations xmlns='http://epoq.de'>

  <recommendationId>f1fecea4-2b20-44b6-b7e4-082fe7005c18</recommendationId>
  <recommendation>
    <productId>REC_PRODUKT_ID1</productId>
    <url><![CDATA[http://rs1.epoq.de/inbound-servletapi/click?tenantId=IHRE
      EPOQ KENNUNG&sessionId=SESSION_ID&recommendationId=f1fecea4-2b20-44b6-
      b7e4-082fe7005c18&productId=REC_PRODUKT_ID1]]></url>
  </recommendation>
  <recommendation>
```

```

<productId>REC_PRODUKT_ID2</productId>
<url><![CDATA[http://rs1.epoq.de/inbound-servletapi/click?tenantId=IHRE
    EPOQ KENNUNG&sessionId=SESSION_ID&recommendationId=f1fecea4-2b20-44b6-
    b7e4-082fe7005c18&productId=REC_PRODUKT_ID2]]></url>
</recommendation>

...
</recommendations>

```

- widgetTheme=multixml: liefert die Empfehlungen im XML Format zurück. Es können mehrere « domains » angefordert werden. Jede steht für eine Regel

```

<?xml version='1.0' encoding='ISO-8859-1' standalone='yes'?>
<recommendations xmlns='http://epoq.de'>
  <domain no='0' rules='default'>
    <recommendationId>7aa7876d-b306-4fcc-8c82-1d8e8f60be92</recommendationId>
    <recommendation>
      <productId>REC_PRODUKT_ID1</productId>
      <url><![CDATA[http://rs1.epoq.de/inbound-servletapi/click?tenantId=IHRE
        EPOQ KENNUNG&sessionId=SESSION_ID&recommendationId=7aa7876d-b306-4fcc
        -8c82-1d8e8f60be92&productId=REC_PRODUKT_ID1]]></url>
    </recommendation>
    <recommendation>
      <productId>REC_PRODUKT_ID2</productId>
      <url><![CDATA[http://rs1.epoq.de/inbound-servletapi/click?tenantId=IHRE
        EPOQ KENNUNG&sessionId=SESSION_ID&recommendationId=7aa7876d-b306-4fcc
        -8c82-1d8e8f60be92&productId=REC_PRODUKT_ID2]]></url>
    </recommendation>
  </domain>
  <domain no='1' rules='xyz'>
    <recommendationId>123435-b306-4fcc-345435-344552g</recommendationId>
    <recommendation>
      <productId>REC_PRODUKT_ID1_A</productId>
      <url><![CDATA[http://rs1.epoq.de/inbound-servletapi/click?tenantId=IHRE
        EPOQ KENNUNG&sessionId=SESSION_ID&recommendationId=123435-b306-4fcc-345435
        -344552g&productId=REC_PRODUKT_ID1_A]]></url>
    </recommendation>
    <recommendation>
      <productId>REC_PRODUKT_ID2_B</productId>
      <url><![CDATA[http://rs1.epoq.de/inbound-servletapi/click?tenantId=IHRE
        EPOQ KENNUNG&sessionId=SESSION_ID&recommendationId=123435-b306-4fcc-345435
        -344552g&productId=REC_PRODUKT_ID2_B]]></url>
    </recommendation>
  </domain>
</recommendations>

```

### 3.3.1.2. erweitertes Format mit zusätzlichen Attributen aus dem Katalog (JSON oder XML)

- widgetTheme=json2 liefert die Empfehlungen im JSON Format zurück :

Beispiel :

```
jQuery16109371099133865392_1336036472034({"recommendations": {"domain": {
"@no": "0",
"@rules": "VariationFilter",
"recommendationId": {"$": "82810ebe-be72-4fb4-a289-48cabab9dbea"},
"items": {"item": [
{
"@no": "0",
"@id": "123176-25",
"@name": "Damen T-Shirt",
"@link": "http:\\\\meinshop.de\\damen\\bekleidung\\shirts-und-tops\\public-damen-t-shirt--7111\\",
"@image_link": "http:\\\\meinshop.de\\images\\product_images\\211.164x164.jpg",
"@price": "",
"@feedbackurl": "http:\\\\rs1.epoq.de\\inbound-servletapi\\click?tenantId=IHRE EPOQ KENNUNG&sessionId=SESSION_ID&recommendationId=82810ebe-be72-4fb4-a289-48cabab9dbea&productId=123176-25&target=http%3A%2F%2Fmeinshop.de%2Fdamen%2Fbekleidung%2Fshirts-und-tops%2Fpublic-damen-t-shirt--7111"
},
{
"@no": "1",
"@id": "113334-15",
...
}]
}}}) ;
```

- widgetTheme=xml2 liefert erweiterte Attribute im XML Format zurück.

Beispiel :

```
?xml version="1.0" encoding="UTF-8"?>

<recommendations xmlns="http://epoq.de" xmlns:g="http://base.google.com/ns/1.0"
xmlns:e="http://base.google.com/cns/1.0" xmlns:c="http://base.google.com/cns/1.0">

  <domain xmlns="" no="0" rules="rule1,rule2">

    <recommendationId>076b7fa9-1led-4cad-a8a4-8dac4884d4f8</recommendationId>

    <items xmlns="http://epoq.de">

      <item xmlns="" no="0" g:id="303381001" g:name="KHUJO Tamarind lightpink"
g:link="http://www.fshop.com/KHUJO-Tamarind-lightpink/303381001/"
g:image_link="http://www.shop.com/gateTools/scripts/binary.php?BinaryCodeID=973735&amp;NoRefC
heck=1" g:price="69.95" feedbackurl="http://rs1.epoq.de/inbound-
servletapi/click?tenantId=shop&amp;sessionId=manakin*699475591&amp;recommendationId=076b7fa9-
1led-4cad-a8a4-
8dac4884d4f8&amp;productId=303381001&amp;target=http%3A%2F%2Fwww.shop.com%2FKHUJO-shop-
lightpink%2F303381001%2F" />
```

```

    <item xmlns="" no="1" g:id="299665001" g:name="INSIGHT Summer Rose Dress white-
multicolored" g:link="http://www.shop.com/INSIGHT-Summer-Rose-Dress-white-
multicolored/299665001/"
g:image_link="http://www.shop.com/gateTools/scripts/binary.php?BinaryCodeID=989384&NoRefC
heck=1" g:price="79.95" feedbackurl="http://rsl.epoq.de/inbound-
servletapi/click?tenantId=shop&sessionId=manakin*699475591&recommendationId=076b7fa9-
11ed-4cad-a8a4-
8dac4884d4f8&productId=299665001&target=http%3A%2F%2Fwww.shop.com%2FINSIGHT-shop-
Rose-Dress-white-multicolored%2F299665001%2F" />

  </items>

</domain>

</recommendations>

```

### 3.3.2. Klickfeedback der Empfehlungen

Damit epoq RS lernen kann, welche Empfehlungen angeklickt werden, muss die Information an epoq übertragen werden. Hierzu gibt es zwei Möglichkeiten:

- Möglichkeit 1: Sie können die URL nutzen, die in dem `<feedbackurl>` oder `<url>` Feld der Antwort steht (siehe oben). Diese URL kann entweder vom Backend oder von Client-Seite aufgerufen werden.
- Möglichkeit 2: Wenn eine Empfehlung angeklickt wird, auf der angezeigten Produktdetail-Seite, muss der zusätzliche Parameter `epoq_recommendationId` gesetzt werden (mit den Eintrag aus dem `<recommendationId>` Feld aus der Empfehlung).

Beispiel: `epoq_recommendationId = '154c337f-6d20-464b-8ccb-6f863e8332c6'`;

Sollte es möglich sein, ein Produkt direkt von der Empfehlung in den Warenkorb zu legen, muss auf der Warenkorbseite der zusätzliche Parameter `epoq_recommendationId` gesetzt werden.

### 3.3.3. Produktdetailseite

Die Empfehlungen für die Produktdetailseite (Empfehlungen für ein Produkt) werden mit der Funktion `getRecommendations()` bei epoq angefordert mit Angabe des Parameters `&rules=default`.

Die Parameter im Einzelnen können Sie hier einsehen : 3.1

#### 3.3.3.1. REST-API

In das Template sollte der Aufruf `getRecommendations()` eingebaut werden. Folgende Aufrufparameter werden benötigt:

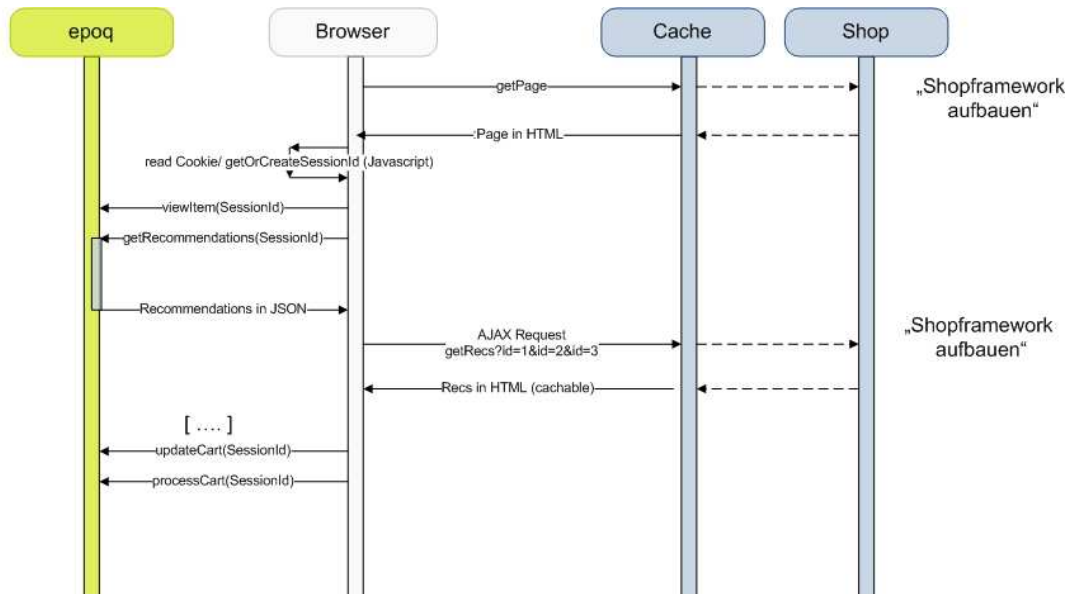
- `tenantId`: "IHRE\_EPOQ\_KENNUNG",
- `sessionId`: "SESSION\_ID",
- `rules`: "default;RULE1;RULE2",
- `productId`: "PRODUKT\_ID",
- `demo`: 6, (nur zu Testzwecken!!)
- `widgetTheme`

### 3.3.3.2. Web-API

Produktempfehlungen können nicht über Web-API eingebunden werden.

### 3.3.3.3. AJAX/JSON

Prozessschaubild:



Anmerkungen:

- Der Shop kennt die epoq SessionId nicht.
- Empfehlungen sind für Suchmaschinen nicht sichtbar.

Um die Empfehlungen über JSON/AJAX anzuzeigen, müssen Sie diese mit folgenden Aufrufparametern abholen:

- tenantId: "IHRE\_EPOQ\_KENNUNG",
- sessionId: "SESSION\_ID",
- rules: "default;RULE1;RULE2",
- productId: "PRODUKT\_ID",
- demo: 6, (nur zu Testzwecken!!)
- widgetTheme

Standardmäßig werden folgende Felder pro Empfehlung zurückgeliefert :

"@no "	Nummer der Empfehlung (0 bis ..)
"@id "	ProduktId
"@name "	Produktname
"@link "	Produkt URL
"@image_link "	Link des Produktbildes (small_Image)
"@price "	Preis (mit zwei Dezimalstellen)
"@feedbackurl "	Feedback URL



Sie können bei dieser Methode auch zusätzliche Felder abfragen. Voraussetzung ist, dass diese Felder im Katalog vorhanden sind und epoq vorliegen. Dazu müssen Sie den Parameter "&fields=FELDDNAME1,FELDDNAME2" übergeben. Zusätzliche Felder müssen von epoq freigegeben werden. Bitte teilen Sie uns in diesem Fall mit, welche Felder Sie benötigen.

Beispielcode für die Anzeige der Empfehlungen (epoqgo.js) :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>AJAX Demo</title>
<script type="text/javascript" src="http://code.jquery.com/jquery-1.6.2.min.js"></script>
<script type="text/javascript" src="http://rs.epoq.de/web-api/epoqgo.js"></script>
</head>

<body>

<p id="epoqWidget1"></p> <!-- here is the recommendations div added-->
<script type="text/javascript">
    //jQuery.noConflict();

    function addRecommendationsToPage(recommendationPids, recommendationId, data)
    {
        alert(recommendationId);

        //customize here
        //=> http://www.IHRE\_URL.com/shop/get-recommendation-htmllsrc.php?pid=4711&pid=0815&pid=1234
        //should deliver html code for the recommendation widget

        var recWidgetFromShopURL = "http://www.IHRE_URL.com/shop/get-recommendation-htmllsrc.php?";
        for(var i=0; i < recommendationPids.length; i++){
            recWidgetFromShopURL+='pid='+recommendationPids[i]+'&';
        }

        //(be careful "same origin policy").

        jQuery.ajax({
            url: recWidgetFromShopURL,
            cache: false,
            success: function(html){
                //alter here for auto tracking clicks on recommendations
                jQuery("#epoqWidget1").append(html);
                jQuery("#epoqWidget1 a").each( function()
                {
                    this.href = this.href + "&egrecqid=" + recommendationId;
                } );

            },
            recommands: {}
        });
    }
};
```

```
}

epoq.go({

  tenantId: "IHRE_EPOQ_KENNUNG",
  sessionId: "SESSION_ID",
  rules: "InCategoryFilter;NotInCategoryFilter",
  productId: "PRODUKT_ID",
  //demo: 6,

  success: function (data)
  {
    var recommendations = data['recommendations'];
    if(typeof(recommendations['domain']) != "undefined")
    {
      //Empfehlungsblock 1
      if(typeof(recommendations['domain'][0]) != "undefined")
      {
        var recommendationPids = new Array();
        for(var i=0; i < recommendations['domain'][0]['items'].item.length; i++)
        {
          recommendationPids[i] = recommendations['domain'][0]['items'].item[i]['@id'];
        }

        var recommendationId = recommendations.domain[0].recommendationId['$'];
        addRecommendationsToPage(recommendationPids, recommendationId, data);

      }
      //Empfehlungsblock 2
      if(typeof(recommendations['domain'][1]) != "undefined")
      {
        var recommendationPids = new Array();
        for(var i=0; i < recommendations['domain'][1]['items'].item.length; i++)
        {
          recommendationPids[i] = recommendations['domain'][1]['items'].item[i]['@id'];
        }

        var recommendationId = recommendations.domain[1].recommendationId['$'];
        addRecommendationsToPage(recommendationPids, recommendationId, data);

      }
    }
  }
});
</script>
</body>
</html>
```

### 3.3.4. Warenkorb

Die Empfehlungen für die Warenkorbseite (Empfehlungen für die Produkte eine Warenkorbes) werden mit der Funktion `getRecommendations()` bei epoq angefordert. Hierfür ist es sehr wichtig, dass die `updateCart()` Funktion (siehe 3.2.2) vor dem `getRecommendations()` Aufruf abgeschickt wird, da epoq die Informationen über den Inhalt des Warenkorbes sonst nicht vorliegen hat. Sie müssen als Parameter folgende Regel übergeben : `&rules=RecommendationForCartRule`

Sie können auch die Warenkorbinformationen bei diesem Aufruf als Parameter übergeben  
(`productId=xxx&price=xxx&quantity=xxx`)

#### 3.3.4.1. REST-API

In das Template sollte der oben beschriebene Aufruf eingebaut werden. Folgende Aufrufparameter werden benötigt:

- `tenantId`: "IHRE\_EPOQ\_KENNUNG",
- `sessionId`: "SESSION\_ID",
- `rules`: "RecommendationForCartRule;RULE1;RULE2",
- `productId`: "PRODUKT\_ID",
- `demo`: 6, (nur zu Testzwecken!!)
- `widgetTheme`

#### 3.3.4.2. Web-API

Warenkorbempfehlungen können nicht über Web-API eingebunden werden.

#### 3.3.4.3. AJAX/JSON

Weitere Details entnehmen Sie aus dem Skript `epoqgo.js` : 3.3.3.3

### 3.3.5. Startseite

Auf der Startseite gibt es verschiedene Arten an Empfehlungen, die Sie anzeigen können. Wir unterscheiden hier zwischen Empfehlungen basierend auf dem kompletten Kundenverhalten und persönlichen Empfehlungen, die auf das individuelle Verhalten eines einzelnen Webshop Besuchers zurückgreifen.

Bei den Empfehlungen, welche auf dem gesamten Klick- und Kaufverhalten basieren, werden in der Regel Topseller geliefert. Um persönliche Empfehlungen anzuzeigen, kann man sich zwischen zwei Verfahren entscheiden :

- Empfehlungen anhand der Kundennummer
- Empfehlungen anhand Cookie basierten Informationen.

Die Empfehlungen anhand der Kundennummer (customerId muss bekannt sein) werden über den Aufruf `getRecommendations()?rules=RecommendationForCustomerRule` abgeholt. Um hier persönliche Empfehlungen anzuzeigen, muss die Kundennummer mit übergeben werden. Dies kann meistens nur erfolgen wenn der Webshop Besucher sich im Shop tatsächlich authentifiziert hat. Dies deckt vermutlich nur einen kleinen Teil der Webshop Besucher ab.

Aus diesem Grund gibt es die Möglichkeit, das Profil eines Kunden in einem Cookie zu speichern. Das Cookie wird unter der Subdomain `<tenantId>.arc.epoq.de` mit dem Namen `eqc_eqe` gespeichert. Der Inhalt des Cookies muss spätestens beim Abholen der Empfehlungen über die Methode `getRecommendations()` als Parameter übergeben werden.

#### 3.3.5.1. REST-API

Startseitenempfehlungen können nicht über REST-API eingebunden werden.

#### 3.3.5.2. Web-API

Startseitenempfehlungen können nicht über Web-API eingebunden werden.

#### 3.3.5.3. AJAX/JSON

Weitere Details entnehmen Sie aus dem Skript `epoq-go.js` : 3.3.3.3

## 4. Best Practices / FAQ

### 4.1. Empfehlungen im Newsletter oder basierend auf Kundendaten

Hier empfehlen wir Ihnen den `viewItem()` Aufruf (Ansicht eines Produkts) über die Web-API zu integrieren.

Falls Sie die Aufrufe aus dem Backend senden (REST-API), muss ein Cookie-Pixel beim Aufruf der `processCart()` Funktion übergeben werden, damit epoq die Daten aus dem Cookie auslesen kann.

### 4.2. Einbau der Empfehlungen

Generell empfehlen wir den `getRecommendations()`-Aufruf aus dem Backend zu senden (REST-API). Die `getRecommendations()`-Aufrufe sind über die Web-API nicht zu empfehlen.

### 4.3. Sonstiges

- In der Regel ist es einfacher die Web-API zu verwenden, bietet aber weniger Möglichkeiten.
- Wenn erhöhte Sicherheit gewünscht wird, kann die SSL Verschlüsselung aktiviert werden.