

# Rtoaster action+ API インターフェース仕様書

2022年4月4日  
 株式会社ブレインパッド

## 改版履歴

作成者	Ver	変更日時	変更履歴
BrainPad	0.1.0	2010/03/24	初版
BrainPad	0.2.0	2010/04/06	<ul style="list-style-type: none"> <li>● 2.2.1.章 AbstractParam クラスのプロパティ appKeys のプロパティ名を appKey に修正</li> <li>● 2.4.章 「300: Internal error」のエラーメッセージを「300: Internal error - Connect Scorer failed」に修正</li> <li>● 3.章 シーケンスを追記</li> </ul>
BrainPad	0.3.0	2010/04/12	<ul style="list-style-type: none"> <li>● 2.3.2.章 SOAP レスポンス (商品自動推奨) のサンプルを追記</li> <li>● 2.3.2.章 SOAP レスポンス (ユーザー個別/自動推奨) のサンプルを追記</li> <li>● 2.4.章 SOAP エラーを追記、修正</li> <li>● 4.2.章 PHP5+SoapClient の SOAP クライアントサンプルを追記</li> </ul>
BrainPad	0.4.0	2010/04/13	<ul style="list-style-type: none"> <li>● 4.2.章 Perl5+SOAP::Lite の SOAP クライアントサンプルを追記</li> </ul>
BrainPad	0.5.0	2010/04/14	<ul style="list-style-type: none"> <li>● 4.2.章 Java+Apache CXF の SOAP クライアントサンプルを追記</li> </ul>
BrainPad	0.6.0	2010/04/19	<ul style="list-style-type: none"> <li>● 2.4.章 エラー「201: Access denied」を追加</li> </ul>
BrainPad	0.7.0	2010/05/25	<ul style="list-style-type: none"> <li>● 2.3.2.章 レコメンド場所 (elementId) に関連するコンテンツが無い場合のレスポンス結果について記載。</li> <li>● 2.3.2.章 「3. SOAP レスポンス (商品自動推奨)」の SOAP レスポンスサンプルを修正</li> <li>● 2.3.2.章 「4. SOAP レスポンス (ユーザー個別/自動推奨)」の SOAP レスポンスサンプルを修正</li> </ul>
BrainPad	0.8.0	2010/05/26	<ul style="list-style-type: none"> <li>● 2.2.1. 章 、 2.2.3. 章 、 2.2.4. 章 AbstractParam クラスと TrackData クラス、RecommendQuery クラスのプロパティ location と referrer について、http:// で始まる URL を指定する必要がある旨を追記。</li> </ul>
BrainPad	0.9.0	2010/05/31	<ul style="list-style-type: none"> <li>● 2.2.1. 章 、 2.2.3. 章 、 2.2.4. 章 AbstractParam クラスと TrackData クラス、RecommendQuery クラスのプロパティ member について、セットする文字列の仕様を明記。</li> </ul>
BrainPad	1.0.0	2010/06/03	<ul style="list-style-type: none"> <li>● 1.6.章 セキュリティを追記</li> <li>● 2.6.章 効果測定 ID の埋込機能を追記</li> </ul>

Rtoaster action+ API インターフェース仕様書  
 (株)ブレインパッド

BrainPad	3.0.0	2010/09/07	<ul style="list-style-type: none"> <li>● 仕様書の Ver を製品プロダクトのリリース番号と統一</li> <li>● 2.6.章 レコメンドコンテンツの差込機能について追記</li> </ul>
BrainPad	3.0.1	2010/12/03	<ul style="list-style-type: none"> <li>● 3.3.章 商品閲覧情報の通知について追記</li> <li>● 3.7.章 非会員から会員への状態変更前後のスコア情報引継について追記</li> </ul>
BrainPad	3.0.5	2011/07/06	<ul style="list-style-type: none"> <li>● SOAP リクエストの XML 書式サンプルの訂正</li> <li>● 効果測定非対応機能についての sessionId の説明を追記</li> </ul>
BrainPad	3.1	2012/03/27	●
BrainPad	3.5	2013/06/11	<ul style="list-style-type: none"> <li>● 2.2.1.章、2.2.4.章 AbstractParam クラスと RecommendQuery クラスにプロパティ userAgent を追加</li> </ul>
BrainPad	4.6	2015/05/19	<ul style="list-style-type: none"> <li>● 2.8 章 追加</li> <li>● 4.1 章 予約語エントリーを追加</li> </ul>
BrainPad	4.6.1	2015/07/08	<ul style="list-style-type: none"> <li>● 2.8 章、3.5 章、4.1 章 誤記を修正</li> </ul>
BrainPad	4.7	2015/09/03	<ul style="list-style-type: none"> <li>● 全体的に文言を修正</li> <li>● 1.5 章 システム概要を修正</li> </ul>
BrainPad	4.8	2017/08/09	<ul style="list-style-type: none"> <li>● 1.3 章に許可 IP アドレスについての文言追加</li> </ul>
BrainPad	5.2	2018/10/31	<ul style="list-style-type: none"> <li>● 2.8 章にアイテム軸自動レコメンドのターゲットアイテム指定方法について記述追加</li> <li>● Jsessionid の保持方法についての記述削除</li> <li>● 各用語の最新化</li> <li>● 2.7 章のフィルタ可能なレコメンド方法について更新</li> </ul>
BrainPad	5.2.1	2018/12/11	<ul style="list-style-type: none"> <li>● 1.3 章の通信許可設定についての文章を修正</li> </ul>
BrainPad	5.3.0	2019/03/04	<ul style="list-style-type: none"> <li>● 2.2 章のパラメータ説明を修正</li> <li>● 2.4 章のエラーコードの追加</li> </ul>
BrainPad	5.8.0	2020/05/22	<ul style="list-style-type: none"> <li>● 4.2 章 Perl をサポート対象から除外し、SOAP クライアントサンプルの記述を削除</li> </ul>

Rtoaster action+ API インターフェース仕様書  
(株)ブレインパッド

BrainPad	5.14	2022/04/04	<ul style="list-style-type: none"><li>● 文言「Rtoaster アカウント ID」を「トラッキング ID」に変更</li><li>● Rtoaster action+内の文言「サイト」を「プロパティ」に変更</li></ul>
----------	------	------------	---

## 目次

1.	はじめに	6
1.1.	対象者	6
1.2.	提供 API	6
1.3.	プロトコル	7
1.4.	文字コード	7
1.5.	システム概要	8
1.6.	セキュリティ	8
2.	インターフェース仕様	9
2.1.	機能一覧	9
2.2.	クラス	10
2.2.1.	AbstractParam	10
2.2.2.	AppKey	10
2.2.3.	TrackData	11
2.2.4.	RecommendQuery	12
2.2.5.	RecommendResult	12
2.3.	メソッド	13
2.3.1.	track メソッド	13
2.3.2.	recommend メソッド	14
2.4.	エラー	18
2.5.	レコメンドコンテンツの種別判定	19
2.6.	レコメンドコンテンツの差し替え機能	20
2.7.	フィルタについて	21
2.8.	アイテム軸自動レコメンド	23
3.	シーケンス	24
3.1.	基本シーケンス	24
3.1.1.	トラッキング	24
3.1.2.	レコメンド	25
3.2.	効果測定(クリック判定)	26
3.3.	商品閲覧情報の通知	27
3.4.	コンバージョン情報の通知	28
3.5.	カートページ等での商品レコメンド	29
3.6.	エリアターゲティング	30
3.7.	非会員から会員への状態変更前後のスコア情報引継	31
4.	付録	32
4.1.	予約語(カスタム変数名)	32
4.2.	SOAP クライアントサンプル	34
4.2.1.	PHP5 + SoapClient	35
4.2.2.	Java + Apache CXF	38

## 1. はじめに

---

本仕様書は、Rtoaster action+ API のインターフェース仕様について記述する仕様書です。

### 1.1. 対象者

本仕様書は、主に Rtoaster action+ API を利用したレコメンデーションのサービスを実装する開発者を対象とする仕様書です。

### 1.2. 提供 API

Rtoaster action+ API では、以下の API を提供します。

API 名	説明
トラッキング API	ユーザーがアクセスしたページの URL・参照元ページの URL 等の情報を Rtoaster action+ にリクエストするための API。スコアリング、自動レコメンドで機械学習を行うために必要となるアイテム (商品) 情報の閲覧並びに購買のログ、効果測定レポートを集計するためのログの蓄積など行う。
レコメンド API	レコメンドを行うための API。ルールベースでは、リクエストされたデータを元に事前に設定されたレコメンドルールを実行し合致したルールのコンテンツ ID 等のレコメンド情報を返す。また自動レコメンドでは、機械学習によってレコメンドされたアイテム (商品) の商品コード等のレコメンド情報を返す。

### 1.3. プロトコル

クライアントとサーバ間は SOAP にてデータ通信を行います。Rtoaster action+が提供する WSDL を元にスタブを生成し、クライアントはスタブを用いて API を呼び出します。

WSDL の URL

<http://rt.rtoaster.jp/api/engine?wsdl>

<https://rt.rtoaster.jp/api/engine?wsdl>

通信プロトコルには HTTP もしくは HTTPS を利用するため、Web サイトのシステム環境 (Web アプリケーションサーバ等) から Rtoaster action+ API Server (rt.rtoaster.jp) に対して、HTTP、HTTPS 通信が許可されている必要があります。

rt.rtoaster.jp は動的に IP アドレスが変わりますので、IP アドレスによるフィルタリングは行えません。

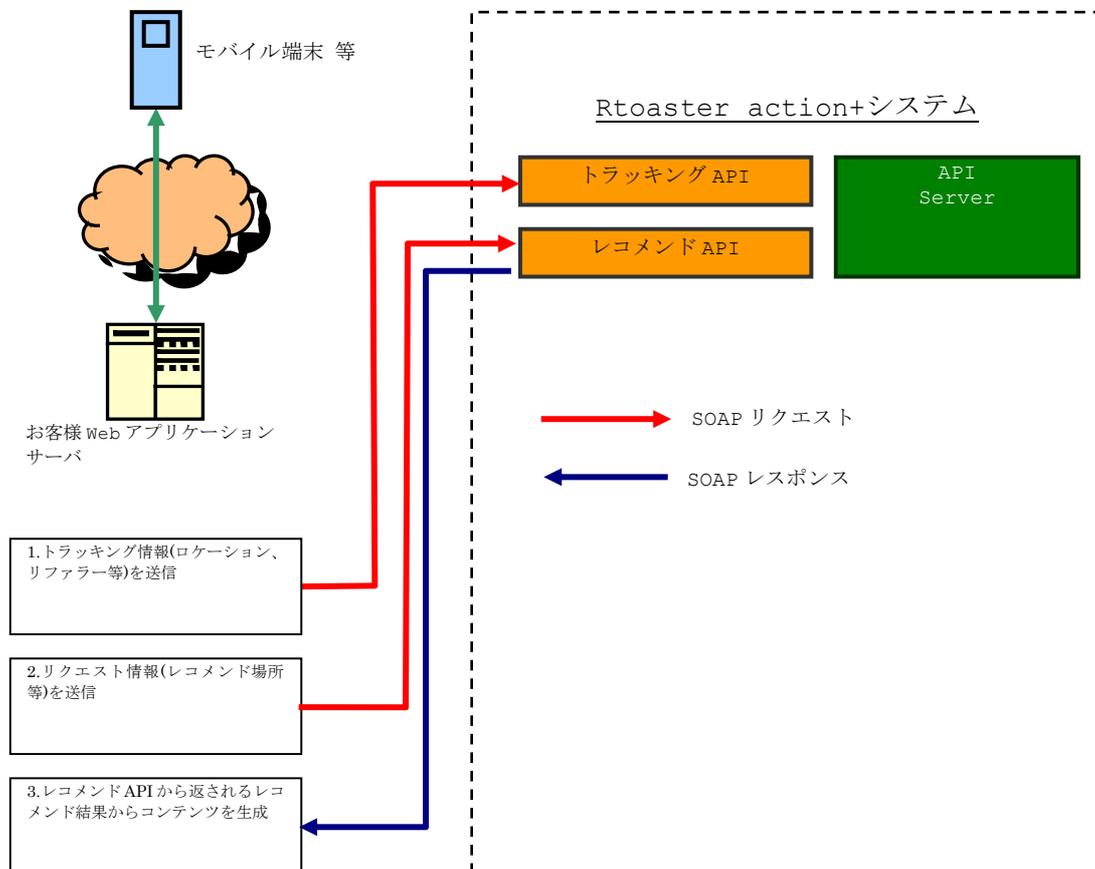
IP アドレスによるフィルタリングが必要な場合は Rtoaster サポートセンターまでご連絡下さい。

### 1.4. 文字コード

API 内部で使われる文字コードは UTF-8 です。API の呼び出し元にて UTF-8 以外の文字コードを扱っている場合には、必要に応じて文字コードの変換処理を行って下さい。

## 1.5. システム概要

Rtoaster action+ API を利用したレコメンドのシステム概要を以下に記載します。



- お客様 Web アプリケーションサーバ上で実行されるアプリケーション (Java、Perl、PHP 等) からトラッキング API・レコメンド API を呼び出し、レコメンド API が返すレコメンド結果を元にお客様 Web アプリケーションサーバ側にてコンテンツを生成します。

## 1.6. セキュリティ

セキュア (SSL) なサイト、ページにてトラッキング API とレコメンド API を実行する場合、Rtoaster action+ API Server とは HTTPS 通信で連携することを推奨します。

**【注意】 HTTP 通信で連携すると平文でデータが送受信されるため、第三者に傍受される危険性があります。**

HTTPS 通信で Rtoaster action+ API Server と連携する方法は、4.2.章 SOAP クライアントサンプルの各サンプルコードを参照して下さい。

## 2. インターフェース仕様

---

本章は、Rtoaster action+ API のインターフェース仕様について記述します。

### 2.1. 機能一覧

Rtoaster action+ API で提供される機能は以下の通りです。

メソッド名	機能
track	トラッキングを実行する
recommend	レコメンドを実行する

トラッキングのみ行うページでは track メソッドを実行し、レコメンドを行うページでは track メソッドを実行した後に recommend メソッドを実行します。

## 2.2. クラス

### 2.2.1. AbstractParam

共通のパラメータをセットするためのクラスです。

クラスのプロパティ	必須	型	説明
accountId	○	String	プロパティ (サイト) ごとに発行される「RTA-」から始まるトラッキング ID (Rtoaster アカウント ID)
member	○	String	サイト訪問者のユニークユーザーID <ul style="list-style-type: none"> <li>● 250 文字以内</li> <li>● 以下の文字列は使用不可 undefined null</li> </ul>
ipAddress	○	String	サイト訪問者の IP アドレス
location	○	String	アクセスされたページの URL <ul style="list-style-type: none"> <li>● http:// または https:// で始まる URL を指定</li> <li>● 8,192 文字以内</li> </ul>
userAgent		String	サイト訪問者のユーザーエージェント情報 <ul style="list-style-type: none"> <li>● 1,024 文字以内</li> </ul>
appKey		List<AppKey>	カスタム変数情報のリスト <ul style="list-style-type: none"> <li>● リストの要素毎に 2 バイトのオーバーヘッドがあります</li> <li>● 合計 65,535 バイト以内</li> </ul>

### 2.2.2. AppKey

カスタム変数の情報をセットするためのクラスです。

クラスのプロパティ	必須	型	説明
name	○	String	カスタム変数名
value	○	String	カスタム変数値

### 2.2.3. TrackData

トラッキングを実行するための情報をセットするクラスです。  
 AbstractParam クラスを継承します。

クラスのプロパティ	必須	型	説明
accountId	○	String	プロパティ (サイト) ごとに発行される「RTA-」から始まるトラッキング ID (Rtoaster アカウント ID)
member	○	String	サイト訪問者のユニークユーザーID <ul style="list-style-type: none"> <li>● 250 文字以内</li> <li>● 以下の文字列は使用不可 undefined null</li> </ul>
ipAddress	○	String	サイト訪問者の IP アドレス
location	○	String	アクセスされたページの URL <ul style="list-style-type: none"> <li>● http:// または https:// で始まる URL を指定</li> <li>● 8,192 文字以内</li> </ul>
userAgent		String	サイト訪問者のユーザーエージェント情報 <ul style="list-style-type: none"> <li>● 1,024 文字以内</li> </ul>
appKey		List<AppKey>	カスタム変数情報のリスト <ul style="list-style-type: none"> <li>● リストの要素毎に 2 バイトのオーバーヘッドがあります</li> <li>● 合計 65,535 バイト以内</li> </ul>
referrer		String	アクセス元ページの URL <ul style="list-style-type: none"> <li>● http:// または https:// で始まる URL を指定</li> <li>● 8,192 文字以内</li> </ul>

- 黄色で色付けしたクラスのプロパティは AbstractParam クラスから継承されるクラスのプロパティ

## 2.2.4. RecommendQuery

レコメンドを実行するための情報をセットするクラスです。  
 AbstractParam クラスを継承しています。

クラスのプロパティ	必須	型	説明
accountId	○	String	プロパティ (サイト) ごとに発行される「RTA-」から始まるトラッキング ID (Rtoaster アカウント ID)
member	○	String	サイト訪問者のユニークユーザー ID <ul style="list-style-type: none"> <li>● 250 文字以内</li> <li>● 以下の文字列は使用不可 undefined null</li> </ul>
ipAddress	○	String	サイト訪問者の IP アドレス
location	○	String	アクセスされたページの URL <ul style="list-style-type: none"> <li>● http:// または https:// で始まる URL を指定</li> <li>● 8,192 文字以内</li> </ul>
userAgent		String	サイト訪問者のユーザーエージェント情報 <ul style="list-style-type: none"> <li>● 1,024 文字以内</li> </ul>
appKey		List<AppKey>	カスタム変数情報のリスト <ul style="list-style-type: none"> <li>● リストの要素毎に 2 バイトのオーバーヘッドがあります</li> <li>● 合計 65,535 バイト以内</li> </ul>
elementId	○	List<String>	レコメンド場所を示すエレメント ID のリスト

- 黄色で色付けしたクラスのプロパティは AbstractParam クラスから継承されるクラスのプロパティ

## 2.2.5. RecommendResult

レコメンドの実行結果がセットされるクラスです。

クラスのプロパティ	必須	型	説明
elementId	○	String	レコメンド場所のエレメント ID
sessionId	○	String	効果測定 ID <ul style="list-style-type: none"> <li>● 効果測定に対応していない機能の場合は値が空の sessionId 属性を出力</li> </ul>
content	○	String	レコメンドコンテンツ

## 2.3. メソッド

### 2.3.1. track メソッド

トラッキングを実行する場合、track メソッドを呼び出します。

形式	void track(TrackData data)
引数	TrackData
戻り値	なし

#### 1. SOAP リクエスト

SOAP リクエストの XML 書式サンプルを以下に記述します。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eng="http://rt.rtoaster.jp/api/engine">
  <soapenv:Header/>
  <soapenv:Body>
    <eng:track>
      <data accountId="RTA-0000-000000">
        <member>member001</member>
        <location>http://localhost/track.html</location>
        <ipAddress>127.0.0.1</ipAddress>
        <!-- Zero or more repetitions: -->
        <appKey name="key1">value1</appKey>
        <appKey name="key2">value2</appKey>
        <!-- Optional1: -->
        <referrer>http://localhost/</referrer>
        <!-- Optional2: -->
        <userAgent>Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2;
Win64; x64; Trident/6.0)</userAgent>
      </data>
    </eng:track>
  </soapenv:Body>
</soapenv:Envelope>
```

## 2.3.2. recommend メソッド

レコメンドを実行する場合、recommend メソッドを呼び出します。

形式	List<RecommendResult> recommend(RecommendQuery query)
引数	RecommendQuery
戻り値	RecommendResult のリスト

- 自動学習のデータが足りない、ルールベースのルールに一致しないなどで Rtoaster action+ 側から返すレコメンドアイテムが無い場合、戻り値として、RecommendResult クラスのリストは返されますが、中身は空 (0 件) のリストが返されます。

### 1. SOAP リクエスト

SOAP リクエストの XML 書式サンプルを以下に記述します。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eng="http://rt.rtoaster.jp/api/engine">
  <soapenv:Header/>
  <soapenv:Body>
    <eng:recommend>
      <query accountId="RTA-0000-000000">
        <member>member001</member>
        <location>http://localhost/test.html</location>
        <ipAddress>127.0.0.1</ipAddress>
        <!-- Zero or more repetitions: -->
        <appKey name="key1">value1</appKey>
        <appKey name="key2">value2</appKey>
        <!-- 1 or more repetitions: -->
        <elementId>element1</elementId>
        <elementId>element2</elementId>
        <!-- Optional: -->
        <userAgent>Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2;
Win64; x64; Trident/6.0)</userAgent>
      </query>
    </eng:recommend>
  </soapenv:Body>
</soapenv:Envelope>
```

※ 本サンプルではオプション項目として、サイト訪問者のユーザーエージェント情報を指定しております。ルール内でサイト訪問者のユーザーエージェントを利用している場合は、SOAP リクエスト内にサイト訪問者のユーザーエージェントが正しく指定されているかご確認ください。

## 2. SOAP レスポンス

SOAP レスポンスの XML 文書サンプルを以下に記述します。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eng="http://rt.rtoaster.jp/api/engine">
  <soapenv:Header/>
  <soapenv:Body>
    <eng:recommendResponse>
      <!-- Zero or more repetitions: -->
      <result elementId="element1"
        sessionId="R0000000000000001">コンテンツ 1</result>
      <result elementId="element2"
        sessionId="R0000000000000002">
        &lt;h1&gt;コンテンツ 2&lt;/h1&gt;</result>
    </eng:recommendResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

### 3. SOAP レスポンス (自動レコメンド)

SOAP レスポンス (自動レコメンド) の XML 文書サンプルを以下に記述します。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eng="http://rt.rtoaster.jp/api/engine">
  <soapenv:Header/>
  <soapenv:Body>
    <eng:recommendResponse>
      <!-- Zero or more repetitions: -->
      <result elementId="item_recommend_1"
        sessionId="T0000000000000001">推奨商品 1</result>
      <result elementId="item_recommend_2"
        sessionId="T0000000000000002">推奨商品 2</result>
      <result elementId="item_recommend_3"
        sessionId="T0000000000000003">推奨商品 3</result>
    </eng:recommendResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP リクエストで指定された elementId が item\_recommend の場合、レコメンド結果の elementId は item\_recommend [1~n] になります。

レコメンド結果は、管理画面の『Web レコメンド設定』で設定する表示枠数分 (最大)、返されます。

#### 4. SOAP レスポンス (ユーザー軸外部リスト/ユーザー軸自動)

SOAP レスポンス (ユーザー軸外部リスト/アイテム軸自動) の XML 文書サンプルを以下に記述します。

- レコメンド場所の表示枠数が未設定

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eng="http://rt.rtoaster.jp/api/engine">
  <soapenv:Header/>
  <soapenv:Body>
    <eng:recommendResponse>
      <!-- Zero or more repetitions: -->
      <result elementId="user_recommend"
        sessionId="P0000000000000001">推奨商品 1</result>
    </eng:recommendResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP リクエストで指定された elementId が user\_recommend の場合、レスポンス結果の elementId は user\_recommend となり、推奨アイテムの情報が 1 件返されます。

- レコメンド場所の表示枠数を設定

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eng="http://rt.rtoaster.jp/api/engine">
  <soapenv:Header/>
  <soapenv:Body>
    <eng:recommendResponse>
      <!-- Zero or more repetitions: -->
      <result elementId="user_recommend_1"
        sessionId="P0000000000000001">推奨商品 1</result>
      <result elementId="user_recommend_2"
        sessionId="P0000000000000002">推奨商品 2</result>
      <result elementId="user_recommend_3"
        sessionId="P0000000000000003">推奨商品 3</result>
    </eng:recommendResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP リクエストで指定された elementId が user\_recommend の場合、レコメンド結果の elementId は user\_recommend [1~n] になります。

レコメンド結果は、管理画面の『レコメンドルール→レコメンドルール設定→レコメンド場所の編集』で設定されている表示枠数分(最大)、返されます。

## 2.4. エラー

SOAP メソッドの呼び出しでエラーが発生した場合、faultstring エレメントでエラーの内容が通知されます。

通知されるエラーは以下の通りです。

エラーメッセージ	説明
100: Invalid parameter	無効なパラメータが指定されました
101: Invalid accountId	無効なトラッキング ID (Rtoaster アカウント ID) です
102: Invalid member	無効なユニークユーザー ID です
103: Invalid ipAddress	無効な IP アドレスです
104: Invalid location	無効な location パラメータです
105: Invalid elementId	無効なエレメント ID です
106: Invalid referrer	無効な referrer パラメータです
107: Invalid userAgent	無効な userAgent パラメータです
108: Invalid appKeys	無効なカスタム変数です
200: License expired	Rtoaster action+のライセンスが切れています
201: Access denied	API へのアクセスが許可されていないホストからのリクエストです
300: Internal error - Tracking failed	トラッキングエラーが発生しました
300: Internal error - Recommend failed	レコメンドエラーが発生しました

### 1. SOAP レスポンス

SOAP レスポンスの XML 文書サンプルを以下に記述します。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eng="http://rt.rtoaster.jp/api/engine">
  <soapenv:Header/>
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>100: Invalid parameter</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

## 2.5. レコメンドコンテンツの種別判定

recommend メソッドのレスポンスで返されるコンテンツの種別は、効果測定 ID の先頭 1 文字で判定します。

種別	説明
D	デフォルトコンテンツ
I	ルールベース (ルール用リスト)
P	ユーザー軸外部リスト・ユーザー軸自動
R	ルールベース (Web レコメンドコンテンツ)
T	アイテム軸外部リスト、アイテム軸自動

## 2.6. レコメンドコンテンツの差し替え機能

Rtoaster action+ API 版の機能として、recommend メソッドのレスポンスで返されるコンテンツの差し替え機能があります。

以下に挙げるプレースホルダをレコメンドコンテンツ、アイテムレコメンドのテンプレート内に記述すると、レスポンス時にプレースホルダが以下の内容で書き換わります。

プレースホルダ	説明
<code>\${_rt.sid}</code>	効果測定 ID に差し替え
<code>\${_rt.uid}</code>	ユーザーID に差し替え
<code>\${カスタム変数名}</code>	カスタム変数の値に差し替え

レコメンドコンテンツ、テンプレート内の任意の場所にプレースホルダを記述します。

```
<a href="/foo/bar/?_rt.sid=${_rt.sid}"/>foo/bar/のページ</a>
```

効果測定 ID が『R0000000001』の場合、以下の様に差し替え、コンテンツをレスポンスします。

```
<a href="/foo/bar/?_rt.sid=R0000000001"/>foo/bar/のページ</a>
```

Recommend メソッドの引数で指定されたカスタム変数の値に差し替えることも可能です。

カスタム変数名 customval1  
カスタム変数値 ABC12345

レコメンドコンテンツ、テンプレート内の任意の場所にプレースホルダを記述します。

```
<a href="/foo/bar/?customval1=${customval1}"/>foo/bar/のページ</a>
```

以下の様に差し替え、コンテンツをレスポンスします。

```
<a href="/foo/bar/?customval1=ABC12345"/>foo/bar/のページ</a>
```

※ レコメンドコンテンツのコンテンツ設定、アイテムレコメンドのテンプレート設定について、Rtoaster サポートセンターに掲載中のマニュアルをご参照ください。

## 2.7. フィルタについて

ユーザーや Web ページのコンテキストに応じてレコメンドの内容を動的に絞り込むフィルタを実装することができます。例えば、ユーザーのしている Web ページの内容や検索キーワード等に応じて、カテゴリを絞ってレコメンドを出したいということが考えられます。このようなフィルタは、Rtoaster action+管理画面の「Web レコメンド設定」において当該レコメンド場所のカテゴリフィルタを有効にし、またレコメンド API を呼び出す際にカスタム変数に当該カテゴリを指定することで可能となります。

それぞれのカスタム変数名と説明について下記テーブルを参照してください。

カスタム変数名	説明	適用可能なレコメンド方法
price_from	価格下限。管理画面上で価格フィルタが有効な場合、この値以上の価格をもつアイテムのみレコメンドされます。	ユーザー軸自動 アイテム軸自動 ユーザー軸外部リスト 閲覧履歴に基づくリアルタイムレコメンド 残留カート内履歴に基づく自動レコメンド 閲覧/購買ランキング 新着アイテム
price_to	価格上限。管理画面上で価格フィルタが有効な場合、この値以下の価格をもつアイテムのみレコメンドされます。	ユーザー軸自動 アイテム軸自動 ユーザー軸外部リスト 閲覧履歴に基づくリアルタイムレコメンド 残留カート内履歴に基づく自動レコメンド 閲覧/購買ランキング 新着アイテム
category1	カテゴリフィルタ (1 階層目)。管理画面上の設定に応じて、指定されたカテゴリのみに絞ったり、別カテゴリを優先したりすることができます。	ユーザー軸自動 ユーザー軸外部リスト 閲覧履歴に基づくリアルタイムレコメンド 残留カート内履歴に基づく自動レコメンド 閲覧/購買ランキング 新着アイテム
category2	カテゴリフィルタ (2 階層目)。管理画面上の設定に応じて、指定されたカテゴリのみに絞ったり、別カテゴリを優先したりすることができます。	ユーザー軸自動 ユーザー軸外部リスト 閲覧履歴に基づくリアルタイムレコメンド 残留カート内履歴に基づく自動レコメンド 閲覧/購買ランキング 新着アイテム

Rtoaster action+ API インターフェース仕様書  
 (株)ブレインパッド

category3	カテゴリフィルタ (3 階層目)。管理画面上の設定に応じて、指定されたカテゴリのみに絞ったり、別カテゴリを優先したりすることができます。	ユーザー軸自動 ユーザー軸外部リスト 閲覧履歴に基づくリアルタイムレコメンド 残留カート内履歴に基づく自動レコメンド 閲覧/購買ランキング 新着アイテム
category4	カテゴリフィルタ (4 階層目)。管理画面上の設定に応じて、指定されたカテゴリのみに絞ったり、別カテゴリを優先したりすることができます。	ユーザー軸自動 ユーザー軸外部リスト 閲覧履歴に基づくリアルタイムレコメンド 残留カート内履歴に基づく自動レコメンド 閲覧/購買ランキング 新着アイテム
category5	カテゴリフィルタ (5 階層目)。管理画面上の設定に応じて、指定されたカテゴリのみに絞ったり、別カテゴリを優先したりすることができます。	ユーザー軸自動 ユーザー軸外部リスト 閲覧履歴に基づくリアルタイムレコメンド 残留カート内履歴に基づく自動レコメンド 閲覧/購買ランキング 新着アイテム

## 2.8. アイテム軸自動レコメンド

「この商品を見た人はこんな商品も見ています」というタイプのレコメンドを実現するためには、閲覧中のアイテム（ターゲットアイテム）のアイテムコードを Rtoaster action+ に通知する必要があります。

これは通常、Rtoaster action+管理画面で設定されたアイテムコード取得正規表現によって商品詳細ページの URL から取得されますが、URL からアイテムコードが取得出来ないサイトではカスタム変数をレコメンド API の呼び出し時に指定する必要があります。

カスタム変数名	説明	適用可能なレコメンド方法
item_code	閲覧された商品のアイテムコード	アイテム軸自動

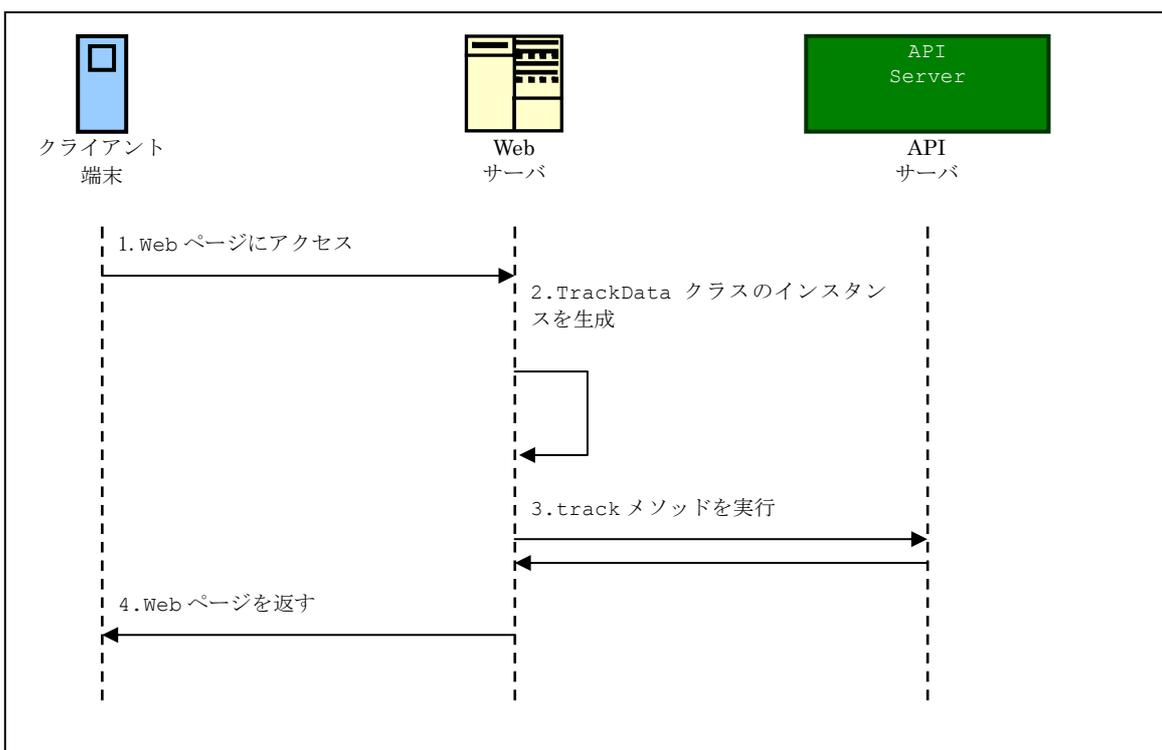
## 3. シーケンス

本章では、クライアント端末、Web サーバ、Rtoaster action+ API サーバ間の処理シーケンスについて記述します。

### 3.1. 基本シーケンス

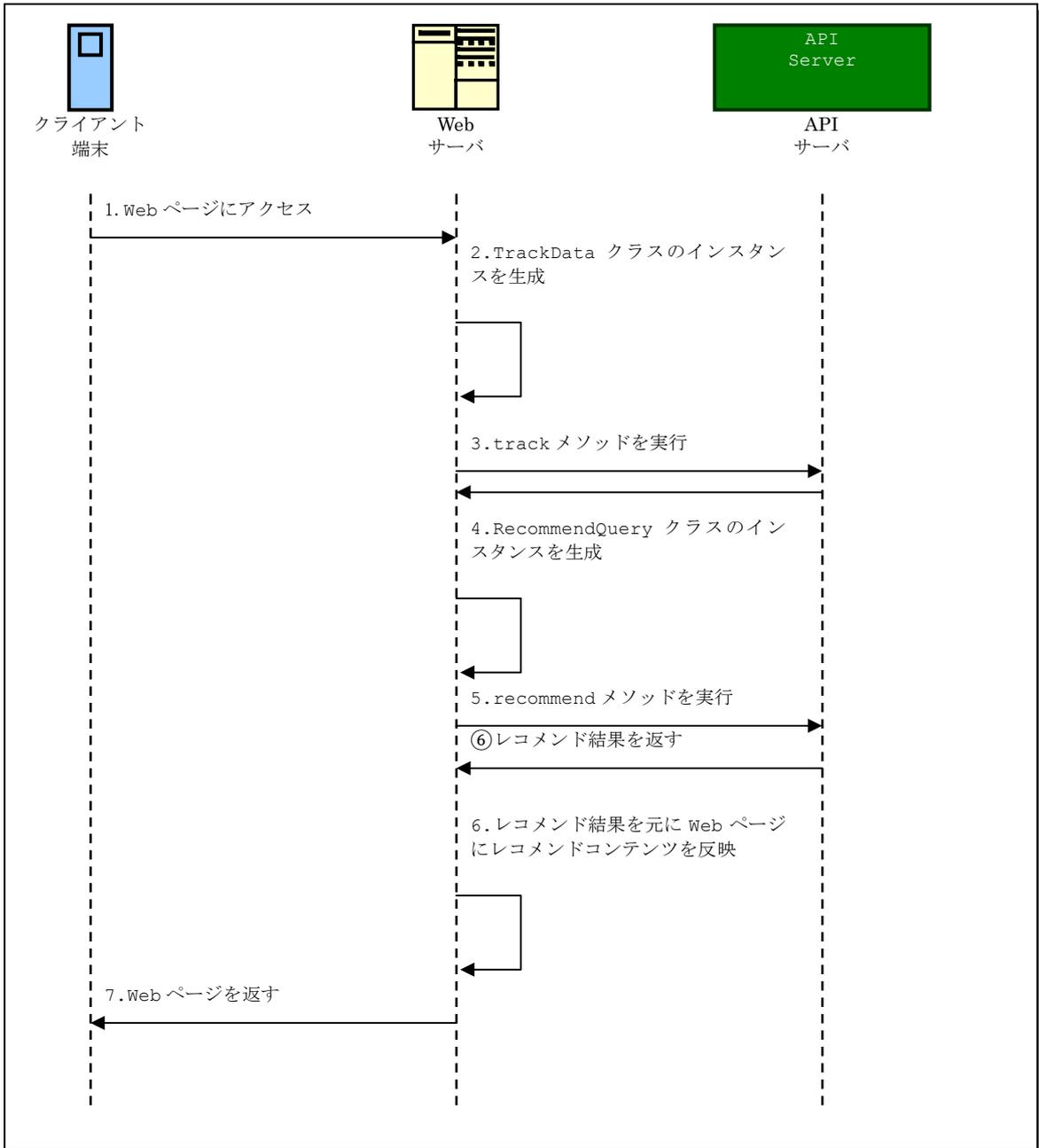
#### 3.1.1. トラッキング

トラッキングのみを行うページでは、track メソッドを実行します。



### 3.1.2. レコメンド

レコメンドを行うページでは、track メソッドを実行した後、recommend メソッドを実行します。

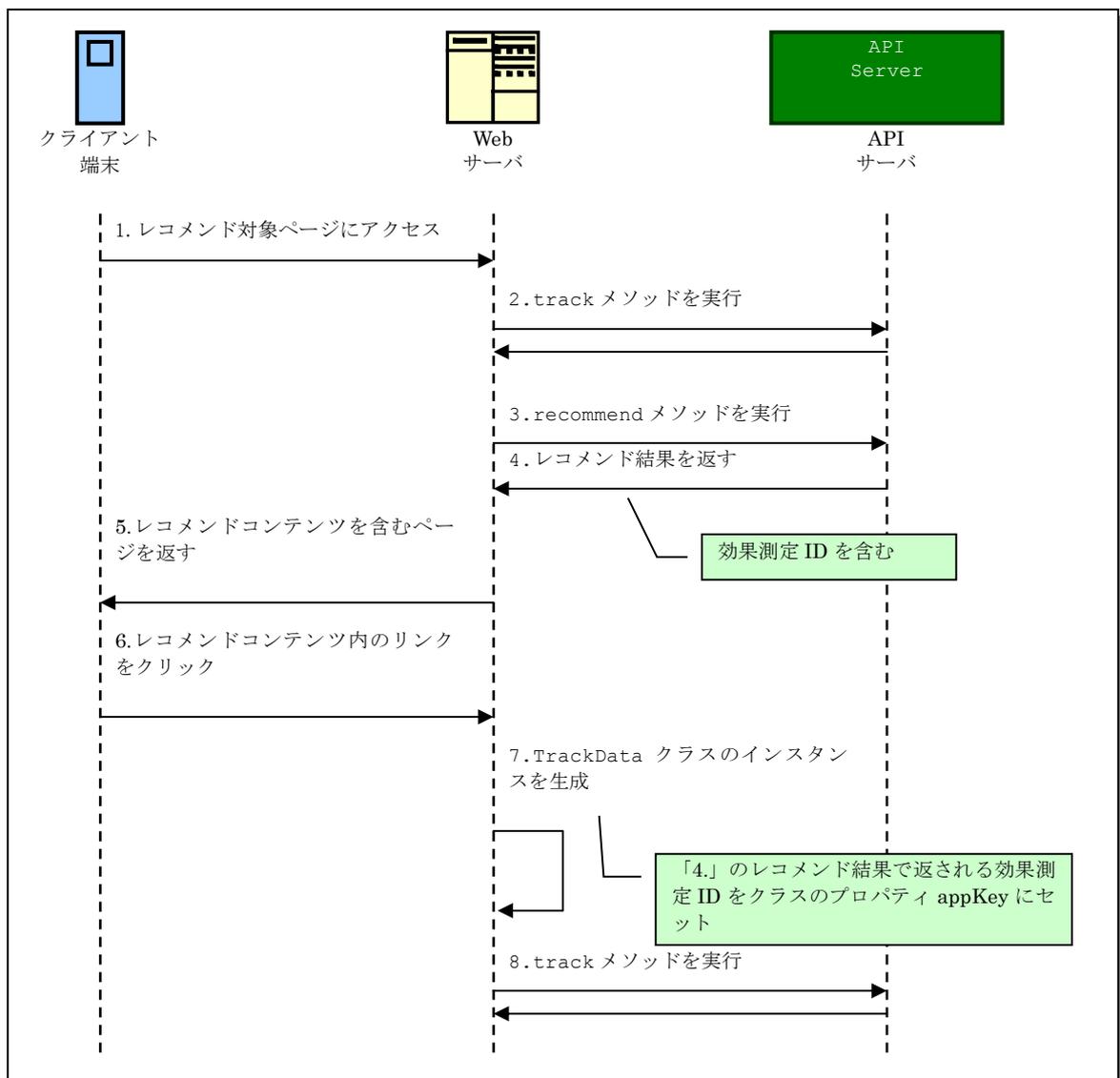


### 3.2. 効果測定(クリック判定)

レコメンドコンテンツの効果測定(クリック判定)を行う場合、レコメンドコンテンツがクリックされ遷移先のページが呼び出されたタイミングで、recommend メソッドのレコメンド結果として返される効果測定 ID を Rtoaster action+ に通知します。

レコメンドコンテンツがクリックされ遷移先のページが呼び出されたタイミングで、TrackData クラスのプロパティ appKey に効果測定 ID をセットし、track メソッドを実行します。

カスタム変数名	必須	説明
_rt.sid	○	recommend メソッドのレスポンスで返される効果測定 ID



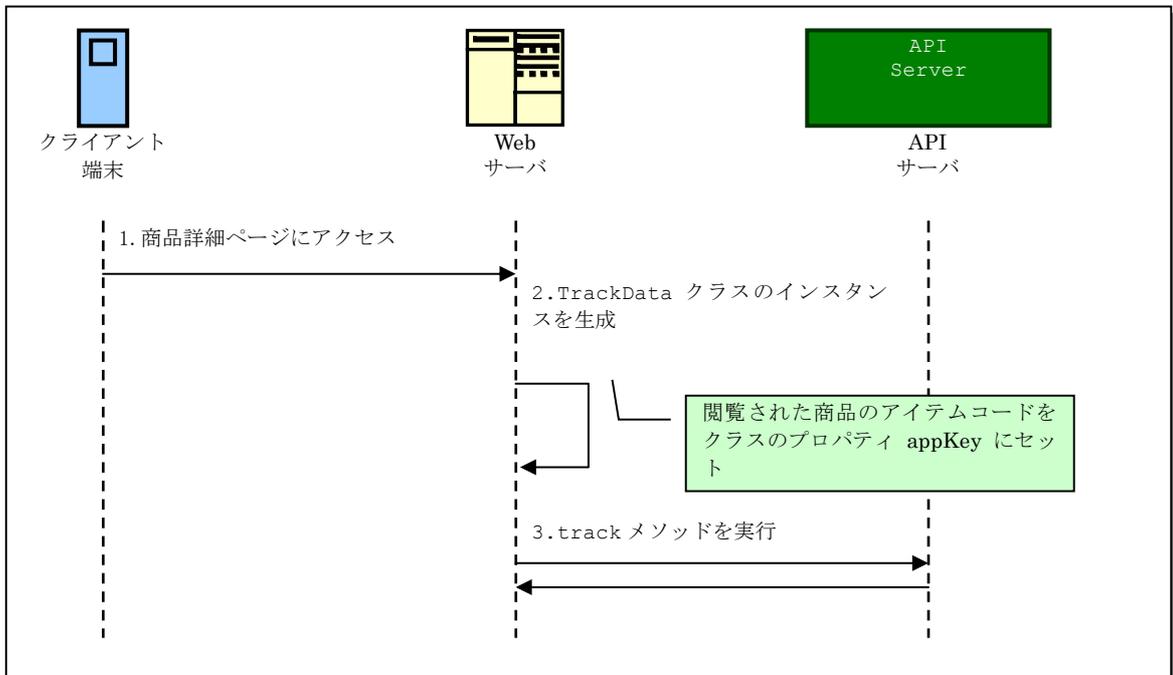
### 3.3. 商品閲覧情報の通知

商品詳細ページが閲覧された際、閲覧された商品のアイテム情報をトラッキング API で Rtoaster action+ に通知します。商品詳細ページの URL からアイテムコードが取得出来ないサイトでこちらの対応が必要となります。

商品詳細ページが呼び出されたタイミングで、TrackData クラスのプロパティ appKey にアイテムコードをセットし、track メソッドを実行します。

カスタム変数名	必須	説明
item_code	○	閲覧された商品のアイテムコード

- カスタム変数名が「item<sub>n</sub>\_code」の場合も閲覧の学習は行われますが、複数のアイテム情報が送信された場合、アイテム情報の閲覧学習は行われません。



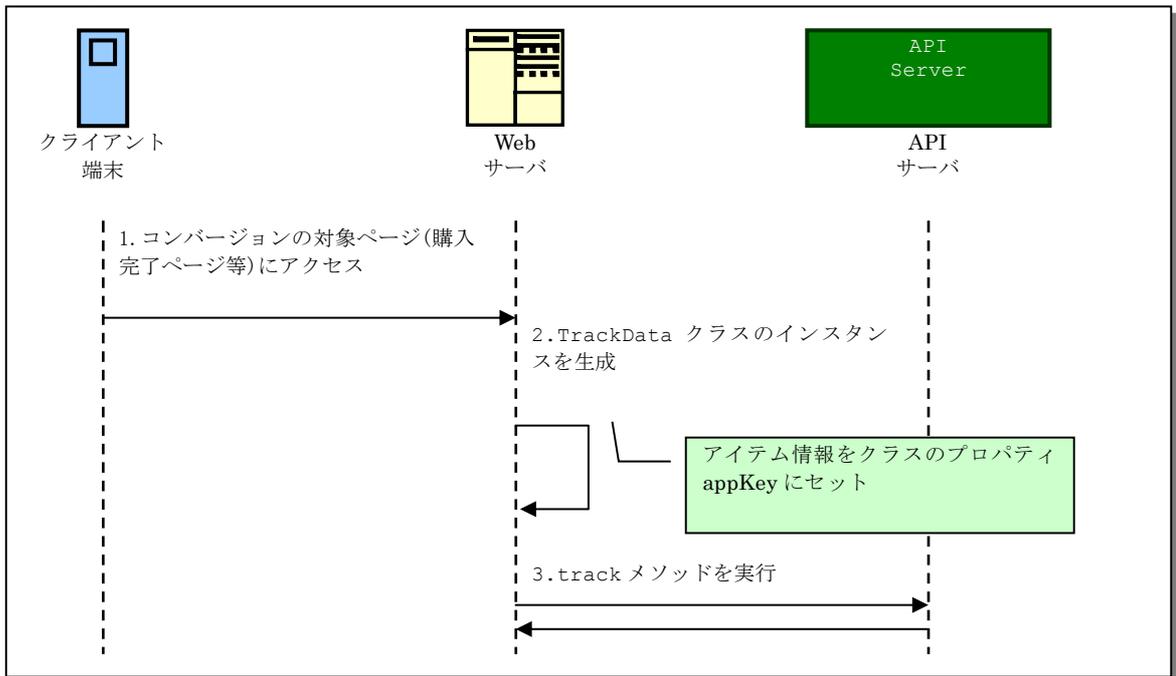
### 3.4. コンバージョン情報の通知

コンバージョン(購買等)の情報に基づいて自動レコメンドの学習を行う場合、コンバージョン時のアイテム情報をRtoaster action+に通知します。

購買完了ページ等、コンバージョン対象のページが呼び出されたタイミングで、TrackDataクラスのプロパティ appKey にアイテム情報をセットし、trackメソッドを実行します。

カスタム変数名	説明
subtotal	小計
item1_code	アイテム1のアイテムコード
item1_price	アイテム1の単価
item1_qty	アイテム1の数量
item2_code	アイテム2のアイテムコード
item2_price	アイテム2の単価
item2_qty	アイテム2の数量
item $n$ _code	アイテム $n$ のアイテムコード
item $n$ _price	アイテム $n$ の単価
item $n$ _qty	アイテム $n$ の数量

- **item1\_code** は必須です。
- 複数のアイテム情報を通知する場合は、全てのアイテム情報をカスタム変数にセットして下さい。

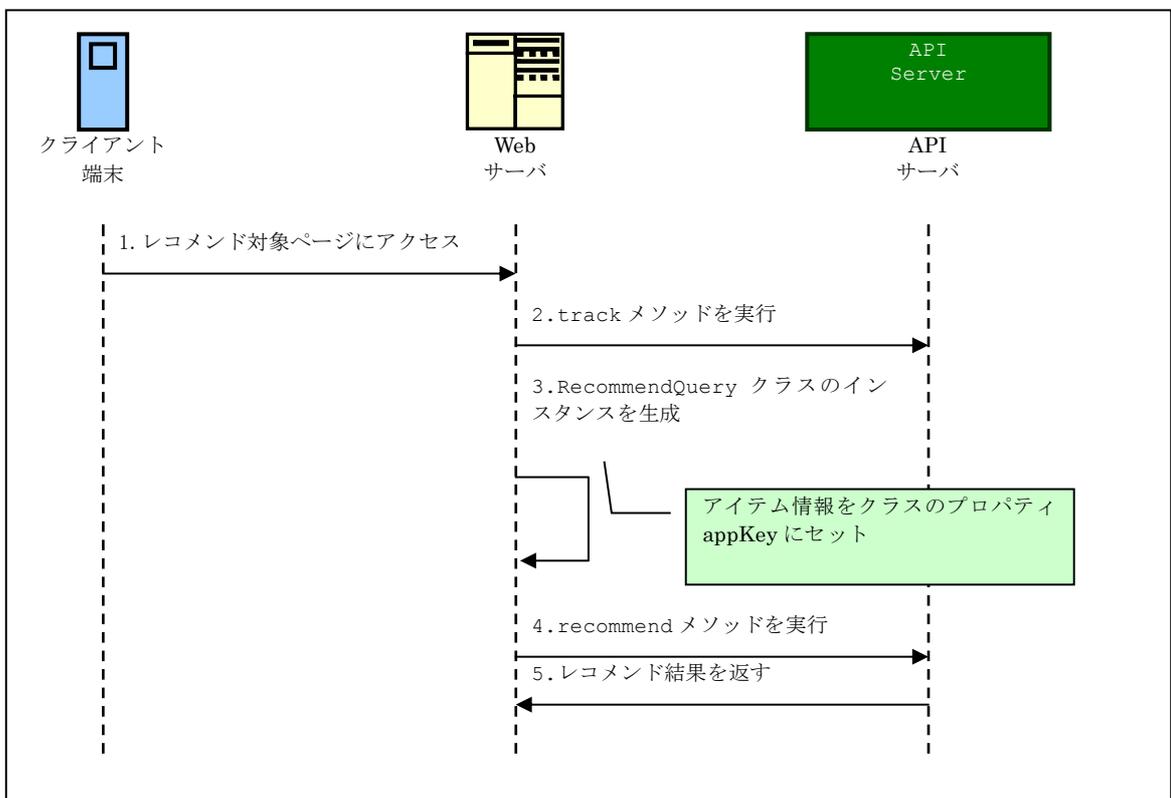


### 3.5. カートページ等での商品レコメンド

カードページなど URL からカテゴリコード、アイテムコードが取得できないページで商品軸のレコメンドを実行する場合、RecommendQuery クラスのプロパティ appKey にアイテム情報をセットして recommend メソッドを実行することにより、商品軸のレコメンドを行うことが可能です。

カスタム変数名	説明
item1_code	アイテム 1 のアイテムコード
item2_code	アイテム 2 のアイテムコード
item $n$ _code	アイテム $n$ のアイテムコード

- **item1\_code** は必須です。
- 複数のアイテム情報を通知する場合は、全てのアイテム情報をカスタム変数にセットして下さい。

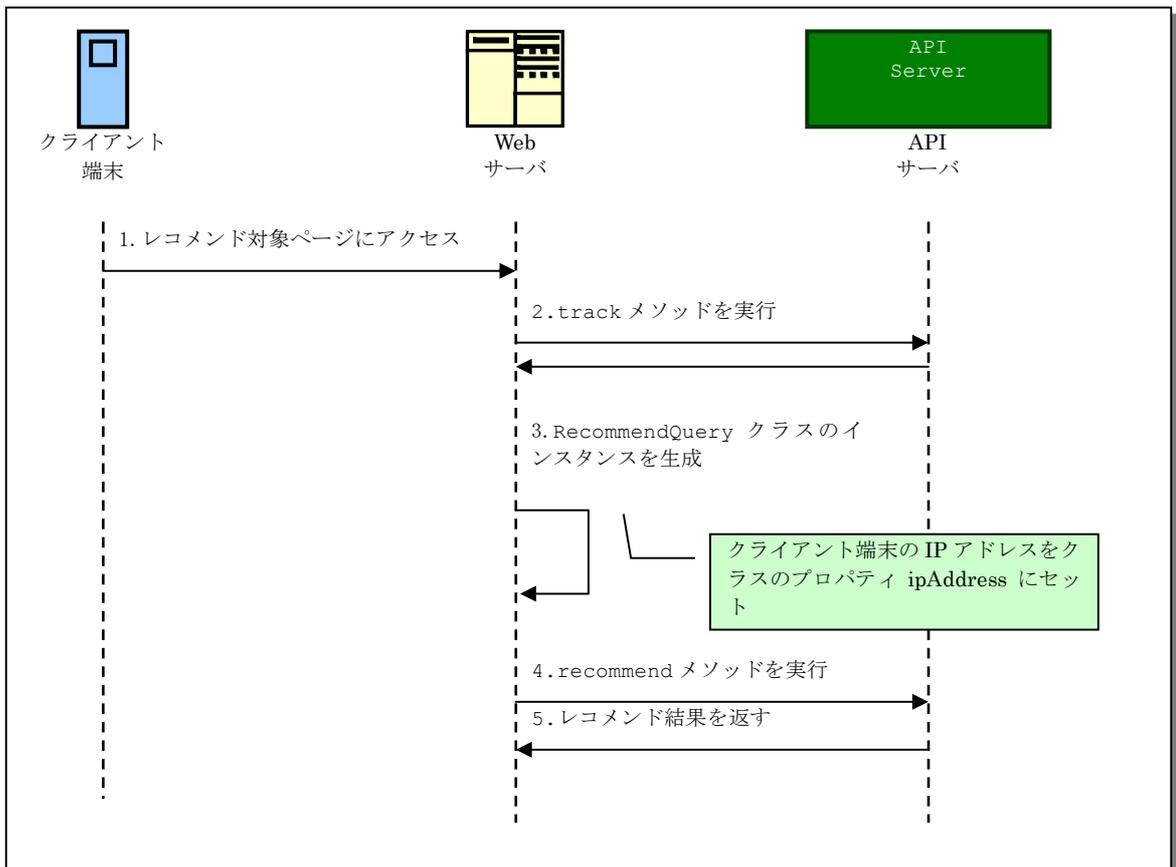


### 3.6. エリアターゲティング

エリアターゲティングによるレコメンドを実行する場合、クライアント端末の IP アドレスを RecommendQuery クラスのプロパティ ipAddress にセットして recommend メソッドを実行することにより、接続元 IP アドレスから地域を判別してレコメンドを行うことが可能です。

※ エリアターゲティングによるレコメンドを行う場合、以下のオプション機能が有効になっている必要があります。

- ・ 国内地域判別
- ・ 海外地域判別
- ・ 企業情報判別

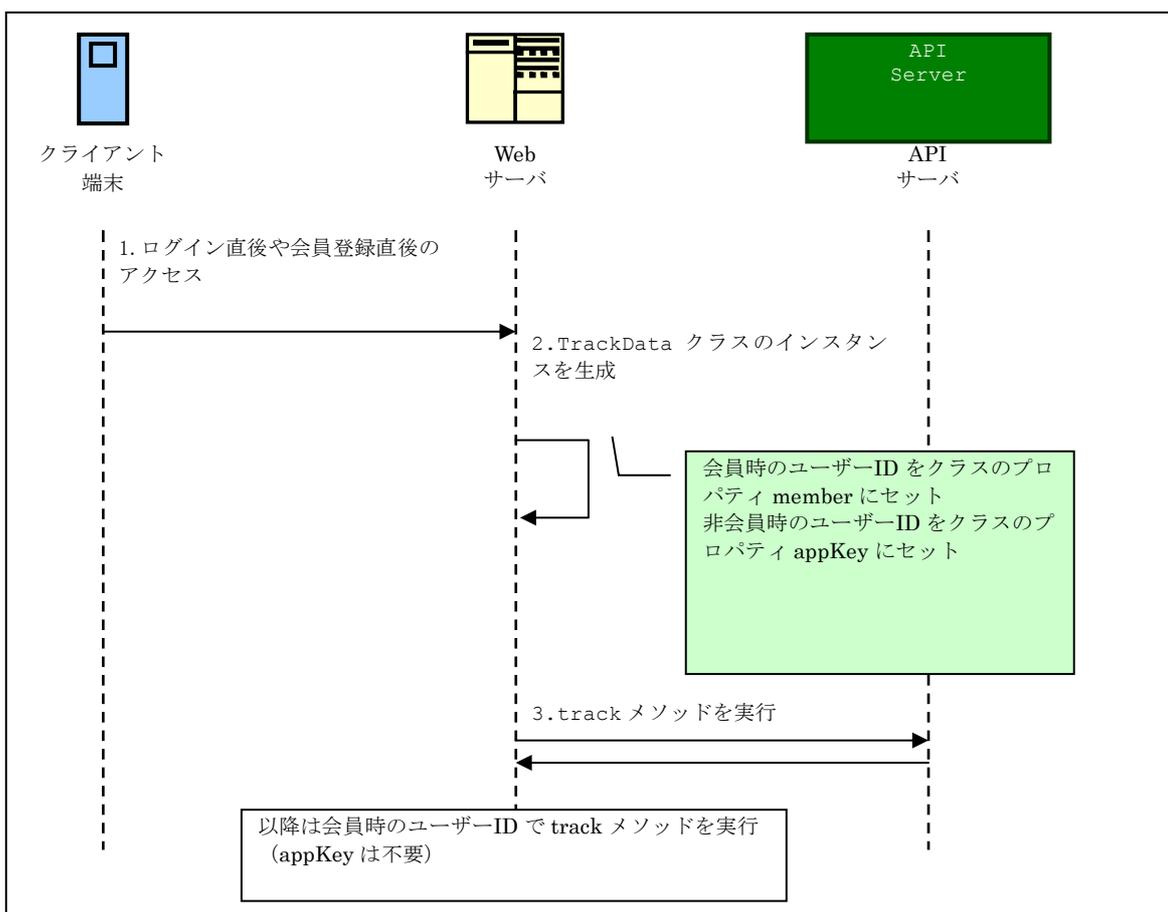


### 3.7. 非会員から会員への状態変更前後のスコア情報引継

ログイン時や会員登録後、非会員状態から会員状態になりユーザーID が変更された場合、通常は状態変化の前後でスコア情報が引き継がれません。ただし、下記の対応を行うことで、会員状態に変更された際、非会員時に付与されたスコア情報を引き継ぐ事が可能です。

- 非会員状態時に発行するユーザーID を RFC 4122 の UUID 形式とする。
- 会員状態時に発行するユーザーID を以下の形式**以外の形式**とする。(※)
  - RFC 4122 の UUID 形式
  - Apache の mod\_usertrack 形式 (<IP address>.<unix time in millis>)**※ユーザーID がいずれかの形式となっていた場合、スコア情報の引継は行われません**
- 会員状態に変更されたタイミング (ログイン時など) で、TrackData クラスのプロパティ appKey に非会員時のユーザーID をセットする。

カスタム変数名	必須	説明
_rt.uid	○	非会員時のユーザーID



## 4. 付録

### 4.1. 予約語(カスタム変数名)

Rtoaster action+では、以下のカスタム変数をレコメンドコンテンツの効果測定・アイテム情報を収集、レコメンド時のフィルタ等の為に利用するため、別の用途で利用することはできません。

カスタム変数名	説明
_rt.sid	効果測定 ID
_rt.uid	ユーザーID
subtotal	小計
item_code	アイテムコード。商品閲覧情報学習で使用。
item $n$ _code	アイテムコード。 $n$ は1から連番
item $n$ _price	単価。 $n$ は1から連番
item $n$ _qty	数量。 $n$ は1から連番
price_from	2.7を参照してください。
price_to	2.7を参照してください。
category1	2.7を参照してください。
category2	2.7を参照してください。
category3	2.7を参照してください。
category4	2.7を参照してください。
category5	2.7を参照してください。
pf	price_from のエリアス
price_f	price_from のエリアス
pfrom	price_from のエリアス
pt	price_to のエリアス
price_t	price_to のエリアス
pto	price_to のエリアス
category	category1 のエリアス
c1	category1 のエリアス
c2	category2 のエリアス
c3	category3 のエリアス
c4	category4 のエリアス
c5	category5 のエリアス

Rtoaster action+ API インターフェース仕様書  
(株)ブレインパッド

grouping	内部的に使用
group	grouping のエリアス
g	grouping のエリアス

## 4.2. SOAP クライアントサンプル

本章では、SOAP クライアントのサンプルを記述します。

**[!]** 注意事項 本章で記述する SOAP クライアントのサンプルは、あらゆる環境において動作を保証するものではありません。

#### 4.2.1. PHP5 + SoapClient

PHP5 環境で SoapClient を利用した SOAP クライアントのサンプルを以下に記述します。

```
<?php
// タイムアウト設定
ini_set('default_socket_timeout', 1);

// クラス定義
class AppKey {
    public $name;
}
class AbstractParam {
    public $accountId;
    public $member;
    public $ipAddress;
    public $location;
    public $userAgent;
    public $appKey = array();
}
class TrackData extends AbstractParam {
    public $referrer;
}
class RecommendQuery extends AbstractParam {
    public $elementId;
}
class RecommendResult {
    public $elementId;
    public $sessionId;
}

// SOAP クライアントのインスタンス生成
// classmap オプションで WSDL 型を PHP クラスにマッピング
$client =
    new SoapClient(
        'http://rt.rtoaster.jp/api/engine?wsdl',
        // セキュアなサイト、ページで API を実行する場合、HTTPS での連携を推奨
        // 'https://rt.rtoaster.jp/api/engine?wsdl',
        array(
            'classmap' =>
                array(
                    'abstractParam' => 'AbstractParam',
                    'appKey' => 'AppKey',
                    'trackData' => 'TrackData',
                    'recommendQuery' => 'RecommendQuery',
                    'recommendResult' => 'RecommendResult'
                ),
            'connection_timeout' => 1
        )
    )
```

```
);

// TrackData クラスのインスタンス生成
$trackData = new TrackData;
$trackData->accountId = 'トラッキング ID (Rtoaster アカウント ID)';
$trackData->member = 'ユニークユーザーID';
$trackData->ipAddress = 'サイト訪問者の IP アドレス';
$trackData->location = 'アクセスされたページの URL';
$trackData->referrer = '参照元ページの URL';
$trackData->userAgent = 'ユーザーエージェント';

$appKey1 = new AppKey();
$appKey1->name = 'カスタム変数名 1';
$appKey1->_ = 'カスタム変数値 1';

$appKey2 = new AppKey();
$appKey2->name = 'カスタム変数名 2';
$appKey2->_ = 'カスタム変数値 2';

$trackData->appKey = array($appKey1, $appKey2);

$track = new StdClass;
$track->data = $trackData;

// track メソッドを実行
try {
    $client->track($track);
} catch (Exception $e) {
    // 例外処理
    var_dump($e->getMessage());
}

// RecommendQuery クラスのインスタンスを生成
$recommendQuery = new RecommendQuery;
$recommendQuery->accountId = 'トラッキング ID (Rtoaster アカウント ID)';
$recommendQuery->member = 'ユニークユーザーID';
$recommendQuery->ipAddress = 'サイト訪問者の IP アドレス';
$recommendQuery->location = 'アクセスされたページの URL';
$recommendQuery->userAgent = 'ユーザーエージェント';
$recommendQuery->elementId = array('レコメンド場所 1', 'レコメンド場所 2');

$appKey3 = new AppKey();
$appKey3->name = 'カスタム変数名 3';
$appKey3->_ = 'カスタム変数値 3';

$appKey4 = new AppKey();
$appKey4->name = 'カスタム変数名 4';
$appKey4->_ = 'カスタム変数値 4';

$recommendQuery->appKey = array($appKey3, $appKey4);
```

```
$recommend = new stdClass;
$recommend->query = $recommendQuery;

// recommend メソッドを実行
try {
    $response = $client->recommend($recommend);
    foreach ($response->result as $result) {
        echo "elementId: " . $result->elementId . "¥n";
        echo "sessionId: " . $result->sessionId . "¥n";
        echo "content: " . $result->_ . "¥n¥n";
    }
} catch (Exception $e) {
    // 例外処理
    var_dump($e->getMessage());
}
?>
```

## 4.2.2. Java + Apache CXF

java 環境で Apache CXF を利用した SOAP クライアントのサンプルを以下に記述します。

まず始めに Apache CXF に同梱される wsdl2java コマンドを利用して、SOAP クライアントの実装に必要なクラスを自動生成します。

```
wsdl2java http://rt.rtoaster.jp/api/engine?wsdl
```

wsdl2java コマンドを実行するとカレントディレクトリ上に、以下の java パッケージが生成されます。

```
jp/rtoaster/rt/api/engine/*  
jp/rtoaster/rt/api/types/*
```

上記のパッケージ内に含まれるクラスを利用し、SOAP クライアントを実装します。

```
import java.util.List;  
import javax.xml.ws.soap.SOAPFaultException;  
import javax.xml.ws.BindingProvider;  
import com.sun.xml.internal.ws.client.BindingProviderProperties;  
import jp.rtoaster.rt.api.engine.*;  
import jp.rtoaster.rt.api.types.*;  
  
public class SoapClient {  
    public static void main(String[] args) {  
        // サービスクラス生成  
        final EngineAPI engine = new EngineAPI();  
        // セキュアなサイト、ページで API を実行する場合、HTTPS 通信での連携を推奨  
        // final Engine API engine = new EngineAPI(new  
        URL("https://rt.rtoaster.jp/api/engine?wsdl"));  
  
        final Engine service = engine.getEnginePort();  
  
        // タイムアウト設定  
        final int timeout = 1000; // ms  
        final Map<String, Object> context =  
(BindingProvider) service). getRequestContext();  
        context.put(BindingProviderProperties.CONNECT_TIMEOUT, timeout); // 接続時の  
        タイムアウト  
        context.put(BindingProviderProperties.REQUEST_TIMEOUT, timeout); // リクエスト  
        時のタイムアウト  
  
        // track メソッドのパラメータ生成  
        final TrackData data = new TrackData();  
        data.setAccountId("トラッキング ID");  
        data.setMember("ユニークユーザーID");  
        data.setLocation("アクセスされたページの URL");  
        data.setIpAddress("サイト訪問者の IP アドレス");  
        data.setReferrer("参照元ページの URL");  
    }  
}
```

```
data.setUserAgent("ユーザーエージェント");

final AppKey appKey1 = new AppKey();
appKey1.setName("カスタム変数名 1");
appKey1.setValue("カスタム変数値 1");

final AppKey appKey2 = new AppKey();
appKey2.setName("カスタム変数名 2");
appKey2.setValue("カスタム変数値 2");

data.getAppKey().add(appKey1);
data.getAppKey().add(appKey2);

// track メソッド実行
try {
    service.track(data);
} catch (SOAPFaultException e) {
    // 例外処理
    System.out.println(e.getFault().getFaultString());
}

// recommend メソッドのパラメータ生成
final RecommendQuery query = new RecommendQuery();
query.setAccountId("トラッキング ID");
query.setMember("ユニークユーザーID");
query.setLocation("アクセスされたページの URL");
query.setIpAddress("サイト訪問者の IP アドレス");
query.setUserAgent("ユーザーエージェント");

final AppKey appKey3 = new AppKey();
appKey3.setName("カスタム変数名 3");
appKey3.setValue("カスタム変数値 3");

final AppKey appKey4 = new AppKey();
appKey4.setName("カスタム変数名 4");
appKey4.setValue("カスタム変数値 4");

query.getAppKey().add(appKey3);
query.getAppKey().add(appKey4);

query.getElementId().add("レコメンド場所 ID1");
query.getElementId().add("レコメンド場所 ID2");

// recommend メソッド実行
try {
    final List<RecommendResult> results = service.recommend(query);
    for (RecommendResult result : results) {
        System.out.println("elementId: " + result.getElementId());
        System.out.println("sessionId: " + result.getSessionId());
    }
}
```

```
        System.out.println("content: " + result.getValue() + "\n");
    }
} catch (SOAPFaultException e) {
    // 例外処理
    System.out.println(e.getFault().getFaultString());
    System.exit(2);
}
}
```