

Tutorial

This short tutorial will guide you through an example TDI usage scenario. It is a good point to start with, as it introduces main TDI principles. It has absolutely no ambitions to present a complete description. For further information, consider reading:

- [Introductory description](#) and motivation behind TDI.
- Full documentation: [Browse side](#) and [protocol description](#).

The tutorial should take only about 5 minutes.

Example Scenario

In this tutorial, we will demonstrate TDI principles on an example e-shop that sells animals. The homepage of such shop may look like this:


Animals of the World

Customer John Doe

Filter






- Two legs
- Four legs
- More legs
- No legs
- Forest
- Grass
- Water
- Air

Purchase



Your shopping cart is empty.

Our Animals

<p>Alligator</p>  <p>Buy \$799</p>	<p>Cat</p>  <p>Buy \$99</p>	<p>Crow</p>  <p>Buy \$79</p>	<p>Dolphin</p>  <p>Buy \$849</p>	<p>Elephant</p>  <p>Buy \$7,999</p>
---	--	---	---	--

1 2 3 4 5 6

We will show some examples of what can TDI do with this page. First, a customer sets a filter, then he or she selects an animal to buy. TDI will be responsible to update the page state using AJAX.

Submit a Form


The home page contains a filter, which is represented by an HTML element that may look like this:

Filter Form Definition

```
<form id="filter-form" method="post" action="/basket/setFilter">
. . .
<input type="checkbox" id="legs-4" name="legs" value="4">
<label for="legs-4">Four legs</label>
. . .
</form>
```

As the first action, the customer sets the filter to show only animals with no legs or more than four legs. After checking the boxes, the filter is submitted:

Animals of the World




Customer John Doe

Filter






<input type="radio"/> Two legs	<input type="radio"/> Forest
<input type="radio"/> Four legs	<input type="radio"/> Grass
<input checked="" type="checkbox"/> More legs	<input type="radio"/> Water
<input checked="" type="checkbox"/> No legs	<input type="radio"/> Air

Purchase



Your shopping cart is empty.

Our Animals

 Alligator Buy \$799	 Cat Buy \$99	 Crow Buy \$79	 Dolphin Buy \$849	 Elephant Buy \$7,999
---	--	---	--	--

Navigation: < 1 2 3 4 5 6 >

Under normal circumstances, the filter settings (values of checkboxes) are sent to the server via a POST request:

Filter Submission (Plain)

```
POST /basket/setFilter HTTP/1.1
Host: ...
. . .
legs=MORE&legs=0
. . .
```

The server is expected to generate a new complete HTML page that will look similarly to the previous one, but with the list of animals updated to reflect the filter state.

TDI in Action

Now let us look what happens when we employ TDI to manage our page. It is done very easily. After including the script in the page source code (see [documentation](#) for details), the only thing you need to do is to add `tdi` class to your form. So it will look somewhat like this:

Filter Form with TDI Enabled

```
<form id="filter-form" method="post" class="tdi" action="/basket/setFilter">
. . .
<input type="checkbox" id="legs-4" name="legs" value="4">
<label for="legs-4">Four legs</label>
. . .
</form>
```

(note the class attribute at the 1st line)

One small change to the page, one giant modification to its behavior. If we submit the form with `tdi` class, TDI script automatically intercepts the action and modifies the request sent to the server:

Filter Submission (TDI)

```
POST /basket/setFilter?_infuse=1&_ts=1380310014837 HTTP/1.1
Host: ...
. . .
legs=MORE&legs=0
. . .
```

(note the extra parameters at the 1st line)

What exactly happened and why? TDI added two extra parameters to the request:

- `_infuse=1` - this tells the server that TDI is in action and the response should be modified accordingly. Namely, instead of returning the whole new page, only a list of TDI instructions is expected in the response.
- `_ts` - contains the current timestamp to avoid caching. Do not bother with it now.

The server sees the `_infuse` parameter, so it only returns the updated list of products (ugh, animals, of course) satisfying new filter settings. To render the new list, the server typically invokes the very same server-side template that is used to generate the list inside a whole page. The response will have `text/xml` content type (instead of `text/html`) and it will look like this:


TDI Update Response

```
<?xml version="1.0"?>
<response>
  <status>OK</status>
  <update target="animal-list">
    <![CDATA[
      . . .
      <li class="product">
        . . .
      </li>
      <div id="animal-list-pagination"> . . . </div>
      . . .
    ]]>
  </update>
</response>
```

Inside the `update` instruction, there is an HTML code corresponding to the new list content. The response tells the client to update the corresponding DOM element.

TDI JavaScript automatically processes the response and performs the update for you. And the page changes without being reloaded:

Animals of the World




Customer
John Doe

Filter





- Two legs
- Four legs
- More legs
- No legs
- Forest
- Grass
- Water
- Air

Purchase



Your shopping cart is empty.

Our Animals

 Buy \$849	 Buy \$0.95	 Buy \$17	 Buy \$139	 Buy \$1.29
--	---	---	--	---

1 2

Going Further

To see a little bit more of TDI features, we will follow our user one more step. Imagine a **Buy** link is clicked:

Animals of the World

Customer John Doe






Filter

- Two-legs
- Four legs
- More legs
- No legs
- Forest
- Grass
- Water
- Air

Purchase

Your shopping cart is empty.

Our Animals

 Buy \$849	 Buy \$0.95	 Buy \$17	 Buy \$139	 Buy \$1.29
--	--	--	--	---

Navigation: < 1 2 >

If we want to process this action with TDI, we just tag the link with `tdi` class again:

Hyperlink with TDI Enabled

```
<a href="/basket/buy?animalId=PY-42" class="green buy tdi">Buy</a>
```

And, once again, TDI intercepts the user action and appends its two extra parameters (`_infuse` and `_ts`) to the request:

Link Followed (TDI)

```
GET /basket/buy?_infuse=1&_ts=1380310014837&animalId=PY-42 HTTP/1.1
Host: ...
...
```

The server does not generate the whole page because of the `_infuse` parameter is present. Instead, it sends necessary updates to the page. In this case, more than one instruction is present, so the logic will have to be a little bit more complicated.

TDI Response with Multiple Instructions

```
<?xml version="1.0"?>
<response>
  <status>OK</status>
  <insert target="basket-total-amount" position="before">
    <![CDATA[
      <li class="one-basket-item">1 Python . . .</li>
    ]]>
  </insert>
  <script id="buy-script" src="js/basket-buy-script.js"></script>
  <update target="user-profile-info" class-add="with-basket" class-remove="empty"
append="true"></update>
  <update target="basket-value">139</update>
  . . .
</response>
```

As a result, the client JavaScript performs the following steps:

- Inserts a new item in the shopping cart, before the total sum.
- Runs a custom JavaScript but only if it has not been run before for this page.
- Remove one CSS class from a user profile box and adds another class. This visually indicates that the shopping cart is non-empty (by a different color, for example).
- Changes the total amount to be paid.
- ... and so on.

After all of the instructions are performed, the page will be updated.

Animals of the World

Customer
John Doe

\$139

Filter

Two legs

Forest

Four legs

Grass

More legs

Water

No legs

Air

Apply Filter

X
Reset

Purchase

1 Python	\$ 139
TOTAL	\$ 139

Check Out

Our Animals

Dolphin

\$849

Fly

\$0.95

Goldfish

\$17

Python

\$139

Worm

\$1.29

◀
1
2
▶

And that's it! Again, the update is done without the page being reloaded.

What else can TDI do for you

The example above was just a quick introduction. TDI can do much more. The concise list of its basic features follows to give you an idea.

- Update the content of some HTML element in the page DOM.
- Add or remove a class of an HTML element.
- Insert a new HTML code to some specified place.
- Run a custom JavaScript. Optionally, it ensures the script is run only once for the page.
- Include a custom style (CSS). Optionally, it ensures the style is used only once for the page.
- Reload the current page completely.
- Redirect the user to a different URL.
- Open a new browser window.
- Also, TDI may be extended with additional custom actions, such as displaying user messages etc.

To provide a feedback to users, TDI allows to add a visual indication of a request being processed. See [Browser-side documentation](#) for details.

If you want to experiment with TDI by yourself you may:

- [Try out TDI Online Demo.](#)
- [Download the demo](#) to your workstation.