



Building dynamic web databases with HTML, JavaScript, PHP & Python

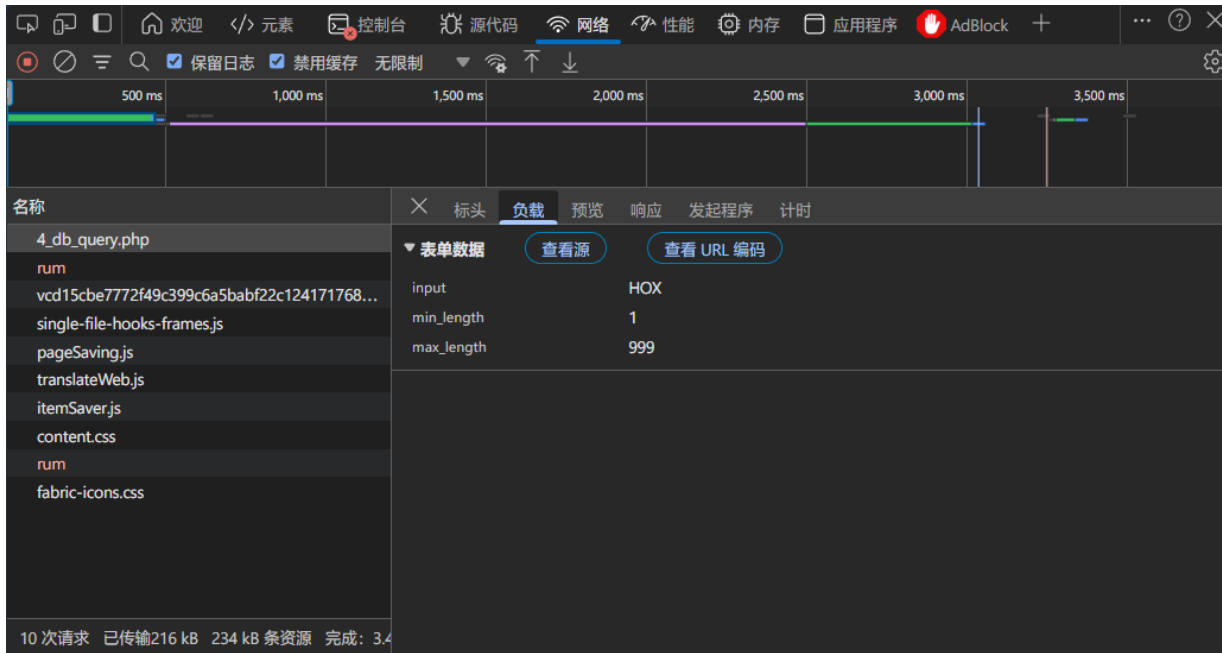
Jiahao Ma, 2025/06/11



A Simple Implementation: tfdb

```
<form action="4_db_query.php" method="post">
  <input type="text" name="input" id="input"><br><br>
  <label for="min_length">Minimum AA Sequence Length: (NO filtering if blank here)</label><br>
  <input type="number" name="min_length" id="min_length"><br><br>
  <label for="max_length">Maximum AA Sequence Length: (NO filtering if blank here)</label><br>
  <input type="number" name="max_length" id="max_length"><br><br>
  <input type="submit" value="Search">
</form>
```

POST this form to "4_db_query.php"
form: input | min_length | max_length





A Simple Implementation: tfdb

4_db_query.php: Query data with parameters

```
$ID = $_POST['input'] ?? ''; // Get parameter from POST request
$minLength = $_POST['min_length'] ?? ''; // Get min length
$maxLength = $_POST['max_length'] ?? ''; // Get max length

// Basic input validation to avoid SQL injection
$ID = mysqli_real_escape_string($conn, $ID);
$minLength = mysqli_real_escape_string($conn, $minLength);
$maxLength = mysqli_real_escape_string($conn, $maxLength);
// Create SQL query
$sql = "SELECT *, LENGTH(seq) FROM tb_factors WHERE (gene_name LIKE '%$ID%' OR gene_id='$ID' OR protein_id LIKE '%$ID%' OR species='$ID' OR family='$ID')";
```

Reading Parameters

Parsing to SQL Format

SQL Query

Present them with echo HTML;

```
// Output each row
while ($row = mysqli_fetch_assoc($result)) {
    echo '<tr>';
    echo '<td>' . htmlspecialchars($row['gene_id']) . '</td>';
    echo '<td>' . htmlspecialchars($row['protein_id']) . '</td>';
    echo '<td>' . htmlspecialchars($row['gene_name']) . '</td>';
    echo '<td>' . htmlspecialchars($row['species']) . '</td>';
    echo '<td>' . htmlspecialchars($row['family']) . '</td>';
    echo '<td>' . htmlspecialchars($row['LENGTH(seq)']) . '</td>';
    echo '</tr>';
}
```

Cons



Deep coupling between **parameters passing, querying and presenting**

Required to design all parts simultaneously;

need to align multiple files when:

- change the parameters list
- change the query method
- change the format of data presenting

Introduction



How to build a dynamic website?

- Basic web page layout (Static text & widget)
- Page control logic (When to do? How to do?)

- Secure data provision
- Efficient data management

Frontend

Backend & Database

An example

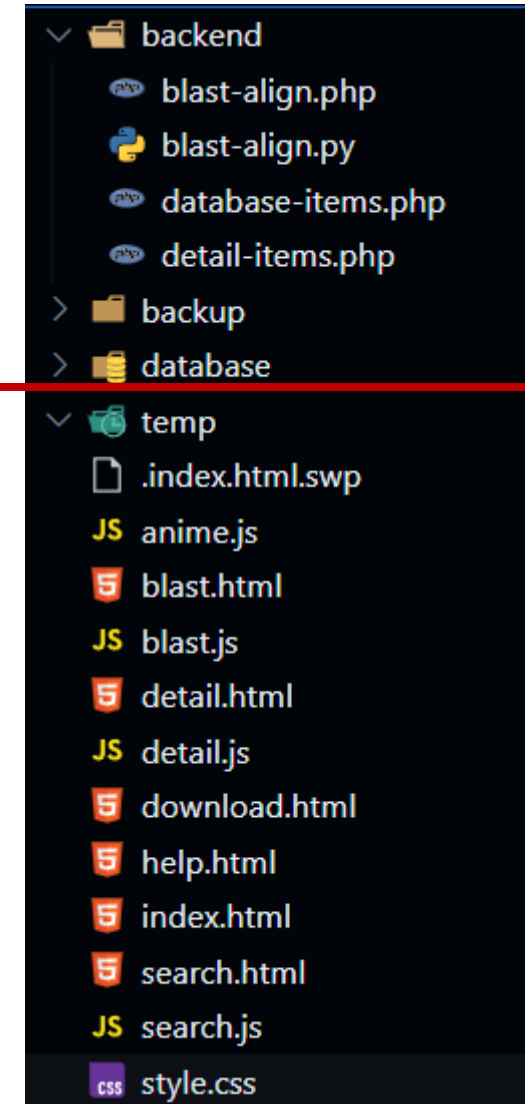
How to build a dynamic website?

- Basic web page layout (Static text & widget)
 - Page control logic (When to do? How to do?)
-
- Secure data provision
 - Efficient data management

Frontend

(Running on client)

Backend & Database
(Running on server)





Basic Frontend-Backend Interaction

./detail.html

```

<header id="top">
  <h2>转录因子详情</h2>
</header>

<div class="content item-detail-container">
  <!-- 详细信息将在此处动态加载 -->
  <div id="itemDetails">
    <p>正在加载详情 ... </p>
  </div>
</div>

```

Replace

./backend/detail-items.php

```

function getConnection($host, $user, $pass, $name) { ...
}

if ($_SERVER['REQUEST_METHOD'] === 'GET') { // 新增: 处理获取详情的请求
  handleGetItemDetail($dbHost, $dbUser, $dbPass, $dbName, $tableName, $tableNameAnno);
} else { ...
}

function handleGetItemDetail($dbHost, $dbUser, $dbPass, $dbName, $tableName, $tableNameAnno) { ...
}

```

Run when loaded

./detail.js

```

document.addEventListener('DOMContentLoaded', () => {
  const itemDetailsContainer = document.getElementById('itemDetails');
  const params = new URLSearchParams(window.location.search);
  const itemId = params.get('id');

  if (!itemId) {
    itemDetailsContainer.innerHTML = '<p>错误: 未提供条目ID。</p>';
    return;
  }

  // 从后端获取特定条目的详细信息
  // 假设后端 API 端点是 /api/database-item/<id>
  async function fetchItemDetails(id) { ...
  }

  function renderItemDetails(item) { ...
  }

  fetchItemDetails(itemId);
});

```

```

// 假设主键列为 'id'
$sql = "select * from {$tableName} inner join {$tableNameAnno} on {$tableName}.gene_id =
{$tableNameAnno}.gene_id where {$tableName}.id = {$itemId}";

$stmt = $conn->prepare($sql);

```

Fetching item details

名称	标头	负载	预览	响应	发起程序	计时
detail-items.php?id=1				1 {		
content.css				"species": "Homo sapiens",		
fabric-icons.css				"protein_id": "ENSP00000386549.3",		
rum				"gene_id": "ENSG00000169953",		
rum				"gene_name": "HSFY2",		
rum				"family": "HSF",		
rum				"seq": "MAHVSSSETQDVSPKDELTAESTRSP",		
detail.html?id=1				"id": 1,		
style.css				"chr": "Y",		
detail.js?v=10.0				"start": 18731440,		
all.min.css				"end": 18773735,		
single-file-hooks-frames.js				"strand": "-",		
pageSaving.js				"gene_symbol": "HSFY2",		
translateWeb.js				"descr": "heat shock transcription f",		
itemSaver.js				"go_bp": "GO:0006357:regulation of t",		
fa-solid-900.woff2				"go_cc": "GO:000785:chromatin;GO:00",		
vcd15cbe7772f49c399c6a5babf22c1241717...				"go_mf": "GO:0003700:sequence-specif",		
fa-solid-900.woff2				"ko": "K04264",		
detail-items.php?id=1				"pathway": ";ko04020 Calcium signal",		
content.css				"length": 401		
fabric-icons.css				}		
rum						

Call function to render items

Using JavaScript to Implement Dynamic Logic

./search.html

```
<script type="text/javascript" src="./search.js?v=4.0"></script>
```

./search.js

- Using EventListener to respond to user actions

- DOMContentLoaded :

```
document.addEventListener('DOMContentLoaded', () => {  
    // 页面HTML结构加载完成后执行初始化代码  
    // ...  
});
```

- 点击事件 (click):

```
applyFiltersBtn.addEventListener('click', () => {  
    currentPage = 1; // 应用新筛选时回到第一页  
    fetchData(currentPage, getCurrentFilters());  
});
```

当用户点击“应用筛选”按钮时，会触发 `fetchData` 函数，根据当前筛选条件重新加载数据。类似的事件监听也用于分页按钮、下载按钮等。

- 选择事件 (change):

```
itemsPerPageSelect.addEventListener('change', (event) => {  
    itemsPerPage = parseInt(event.target.value);  
    currentPage = 1; // 更改每页条目数时，重置到第一页  
    fetchData(currentPage, getCurrentFilters());  
});
```

当用户更改“每页显示条目”下拉框的选项时，会更新 `itemsPerPage` 变量并重新获取数据。

Using JavaScript to Implement Dynamic Logic

./search.html

```
<script type="text/javascript" src="./search.js?v=4.0"></script>
```

./search.js

- Using EventListener to respond to user actions
- **HTML DOM Manipulation**

○ **获取元素:**

```
const resultsListContainer = document.getElementById('resultsListContainer');
const filterKeywordInput = document.getElementById('filterKeywordInput');
```

○ **创建和添加元素:**

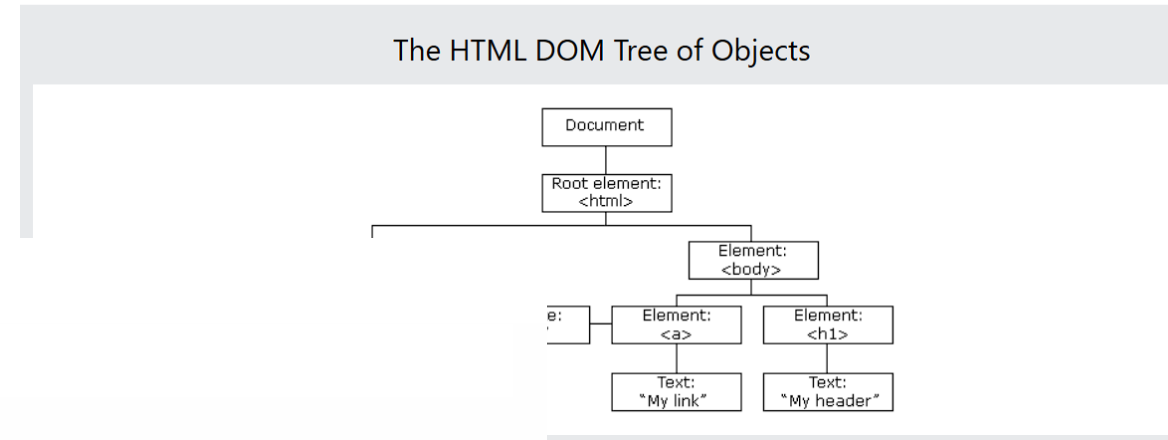
```
// 在 renderResults 函数中
const ul = document.createElement('ul');
items.forEach(item => {
    const li = document.createElement('li');
    // ... 设置 li 的内容和属性 ...
    ul.appendChild(li);
});
resultsListContainer.appendChild(ul);
```

○ **修改内容:**

```
resultsListContainer.innerHTML = resultsListContainer.innerHTML + `
// 或
button.textContent = text`
```

使用 `innerHTML` 或 `textContent` 修改元素的显示内容。例如，在数据加载时显示“正在加载数据...”。

The **HTML DOM** model is constructed as a tree of **Objects**:





Using JavaScript to Implement Dynamic Logic

./search.html

```
<script type="text/javascript" src="./search.js?v=4.0"></script>
```

./search.js

- Using EventListener to respond to user actions
- HTML DOM Manipulation
- **Asynchronous Fetch API**

```
async function fetchData(page = 1, filters = {}) {
  resultsListContainer.innerHTML = '<p>正在加载数据 ... </p>';
  const apiUrl = `./backend/database-items.php?action=query&page=${page}&limit=$

  try {
    const response = await fetch(apiUrl); // 发送GET请求
    if (!response.ok) {
      // 处理HTTP错误
      throw new Error(`HTTP error! status: ${response.status}`);
    }
    const data = await response.json(); // 解析JSON响应
    if (data && data.items) {
      renderResults(data.items);
      renderPagination(data.currentPage || page, data.totalPages);
    } else {
      // 处理数据格式错误
    }
  } catch (error) {
    // 处理网络错误或其他异常
    console.error('加载数据失败:', error);
    resultsListContainer.innerHTML = `<p>加载数据失败: ${error.message}</p>`;
  }
}
```



Using JavaScript to Implement Dynamic Logic

./search.html

```
<script type="text/javascript" src="./search.js?v=4.0"></script>
```

./search.js

- Using EventListener to respond to user actions
- HTML DOM Manipulation
- Asynchronous Fetch API
- **Dynamic rendering results**

```
function renderResults(items) {
  resultsListContainer.innerHTML = ''; // 清空旧结果
  if (!items || items.length === 0) {
    resultsListContainer.innerHTML = '<p>未找到符合条件的条目。</p>';
    // ... 更新全选框状态 ...
    return;
  }
  // ... 启用并显示全选框 ...

  const ul = document.createElement('ul');
  items.forEach(item => {
    const li = document.createElement('li');
    // ... 创建复选框 ...
    // ... 创建链接元素显示条目信息 ...
    li.appendChild(checkbox);
    li.appendChild(link);
    ul.appendChild(li);
  });
  resultsListContainer.appendChild(ul);
  updateSelectAllCheckboxState();
}
```



Basic Frontend-Backend Interaction

./detail.html

```

<header id="top">
  <h2>转录因子详情</h2>
</header>

<div class="content item-detail-container">
  <!-- 详细信息将在此处动态加载 -->
  <div id="itemDetails">
    <p>正在加载详情 ... </p>
  </div>
</div>

```

Replace

./backend/detail-items.php

```

function getConnection($host, $user, $pass, $name) { ...
}

if ($_SERVER['REQUEST_METHOD'] === 'GET') { // 新增: 处理获取详情的请求
  handleGetItemDetail($dbHost, $dbUser, $dbPass, $dbName, $tableName, $tableNameAnno);
} else { ...
}

function handleGetItemDetail($dbHost, $dbUser, $dbPass, $dbName, $tableName, $tableNameAnno) { ...
}

```

Run when loaded

./detail.js

```

document.addEventListener('DOMContentLoaded', () => {
  const itemDetailsContainer = document.getElementById('itemDetails');
  const params = new URLSearchParams(window.location.search);
  const itemId = params.get('id');

  if (!itemId) {
    itemDetailsContainer.innerHTML = '<p>错误: 未提供条目ID。</p>';
    return;
  }

  // 从后端获取特定条目的详细信息
  // 假设后端 API 端点是 /api/database-item/<id>
  async function fetchItemDetails(id) { ...
  }

  function renderItemDetails(item) { ...
  }

  fetchItemDetails(itemId);
});

```

```

// 假设主键列为 'id'
$sql = "select * from {$tableName} inner join {$tableNameAnno} on {$tableName}.gene_id =
{$tableNameAnno}.gene_id where {$tableName}.id = {$itemId}";

$stmt = $conn->prepare($sql);

```

Fetching item details

名称	标头	负载	预览	响应	发起程序	计时
detail-items.php?id=1				1 {		
content.css				"species": "Homo sapiens",		
fabric-icons.css				"protein_id": "ENSP00000386549.3",		
rum				"gene_id": "ENSG00000169953",		
rum				"gene_name": "HSFY2",		
rum				"family": "HSF",		
rum				"seq": "MAHVSSETQDVSPKDELTAESTRSP",		
detail.html?id=1				"id": 1,		
style.css				"chr": "Y",		
detail.js?v=10.0				"start": 18731440,		
all.min.css				"end": 18773735,		
single-file-hooks-frames.js				"strand": "-",		
pageSaving.js				"gene_symbol": "HSFY2",		
translateWeb.js				"descr": "heat shock transcription f",		
itemSaver.js				"go_bp": "GO:0006357:regulation of t",		
fa-solid-900.woff2				"go_cc": "GO:000785:chromatin;GO:00",		
rum				"go_mf": "GO:0003700:sequence-specif",		
rum				"ko": "K04264",		
rum				"pathway": ";ko04020 Calcium signal",		
rum				"length": 401		

Call function to render items