

شما در حال مطالعه نسخه آفلاین یکی از مطالب «مجله فرادرس» هستید. لطفاً توجه داشته باشید، ممکن است برخی از قابلیت‌های تعاملی مطالب، مانند امکان پاسخ به پرسش‌های چهار گزینه‌ای و مشاهده جواب صحیح آن‌ها، نمایش نتیجه آزمون‌ها، پاسخ تشریحی سوالات، پخش فایل‌های صوتی و تصویری و غیره، در این نسخه در دسترس نباشند. برای دسترسی به نسخه آنلاین مطلب، استفاده از کلیه امکانات آن و داشتن تجربه کاربری بهتر [اینجا کلیک کنید](#).

برنامه نویسی

CRUD چیست؟ - از مفهوم تا اجرا به زبان ساده

۱۰۱۶ ۱۴۰۴/۰۳/۲۰ ۱۲ دقیقه

امکان دانلود نسخه PDF

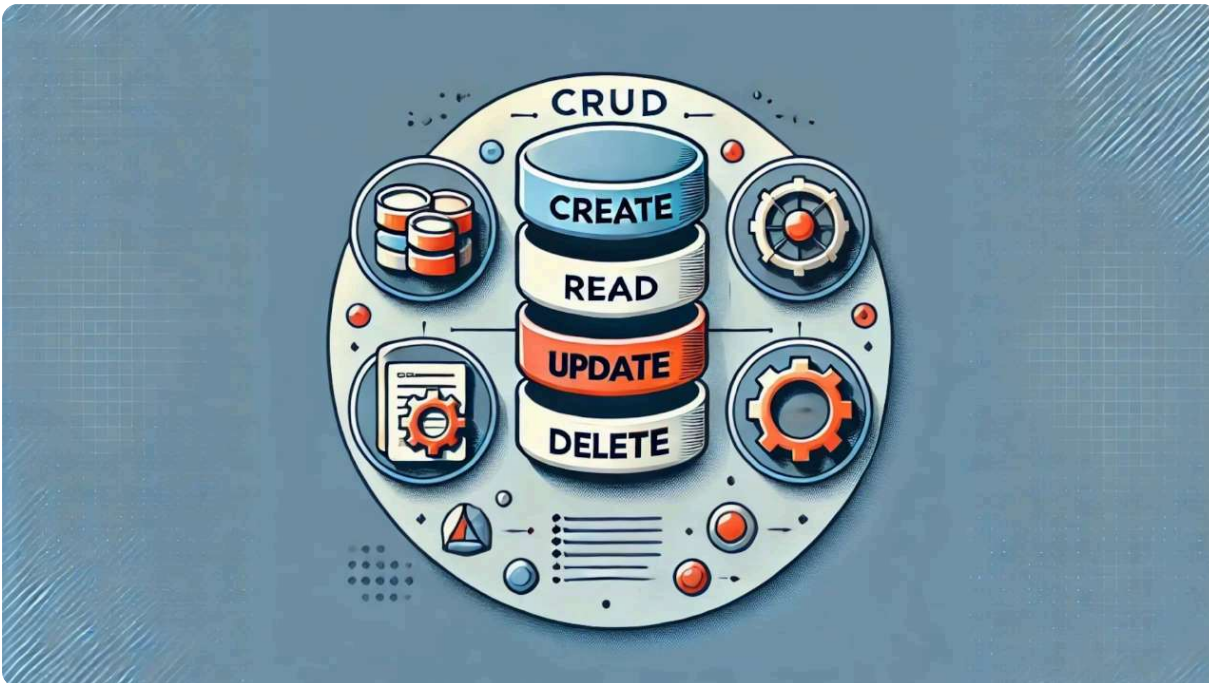
آموزش متنی جامع

عبارت **CRUD** سرنامی از دستورهای **Create** و **Read** و **Update** و **Delete** است. این چهار دستور، توابع پایه‌ای هستند که هر نرم‌افزار کاربردی باید بتواند انجام دهد. در چنین اپلیکیشن‌هایی کاربران باید بتوانند داده ایجاد کنند. برای خواندن داده‌ها با کمک رابط کاربری، دسترسی مورد نیاز را داشته باشند. در صورت نیاز بتوانند داده‌ها را به‌روزرسانی کرده و حتی حذف کنند. اپلیکیشن‌های کاملی که از **CRUD** پشتیبانی می‌کنند شامل سه بخش مهم هستند. این بخش‌ها شامل «**API** و سرور»، «پایگاه داده» و «رابط کاربری» هستند. **API** شامل کدها و متدها است. پایگاه داده اطلاعات کاربر را ذخیره کرده و به فراخوانی این اطلاعات توسط کاربران کمک می‌کند. رابط کاربری هم به کاربران برای تعامل با اپلیکیشن کمک می‌کند. برنامه‌های **CRUD** را با استفاده از هر زبان برنامه‌نویسی می‌توان نوشت. برای اینکه بدانیم رابطه پایگاه داده با **CRUD** چیست، تمام دستورات چهارگانه را یک به یک همراه با نمایش مثال ساده‌ای توضیح داده‌ایم.

▶ آنچه در این مطلب می‌آموزید:

- ✓ یاد می‌گیرید چهار عملیات اصلی **CRUD** را در نرم‌افزارها شناسایی کنید.
- ✓ خواهید آموخت اجرای هر عملیات **CRUD** را در دیتابیس مدیریت کنید.
- ✓ یاد می‌گیرید ارتباط متدهای **HTTP** با عملیات **CRUD** را تحلیل کنید.
- ✓ می‌آموزید چگونه از دستورات **SQL** برای پیاده‌سازی **CRUD** بهره ببرید.

- ✓ با نقش شناسه یکتا (id) در مدیریت داده‌ها آشنا می‌شوید.
- ✓ خواهید توانست کاربرد CRUD را در معماری REST ارزیابی کنید.
- ✓ اهمیت CRUD در مسیر یادگیری برنامه‌نویسی وب را درک می‌کنید.
- ✓ منابع معتبر آموزش CRUD و پایگاه داده را شناسایی خواهید کرد.



فهرست مطالب این نوشته

■ **CRUD چیست؟** (اینجا هستید)

■ تعریف دقیق دستورهای CRUD

Create ■

Read ■

Update ■

Delete ■

■ چگونه با کمک فرادرس طراحی سایت بیاموزیم؟

■ **ارتباط بین CRUD و REST چیست؟**

■ دستور Create در REST

■ دسته Read در REST

نمایش همه

در این مطلب از مجله فرادرس، متوجه می‌شویم که CRUD چیست و از چه دستورهای تشکیل شده است. ابتدا می‌بینیم که در **محیط REST** روش پیاده‌سازی دستورات CRUD چیست. در نهایت هم معادل این دستورات را برای اجرا در پایگاه‌های داده رابطه‌ای نمایش داده‌ایم.

CRUD چیست؟

در زمان ساختن API-ها تمام مدل‌های طراحی شده باید بتوانند ۴ عملیات اصلی را انجام دهند. این عملیات‌ها Create و Read و Update و Delete هستند. در فهرست زیر، توضیح مختصری را برای معرفی عملیات CRUD نوشته‌ایم.

- **Create:** به منابع حافظه مرتبط با مدل، داده‌های جدید بیفزاید.
- **Read:** داده‌های موجود را بخواند.
- **Update:** در صورت نیاز داده‌های مشخص شده را به‌روزرسانی کند.
- **Delete:** داده‌های غیرضروری را از منبع حذف کند.

فیلم آموزش عملیات CRUD در PHP در فرادرس

کلیک کنید

در ادامه مطلب و با هدف اینکه بهتر متوجه شویم CRUD چیست، تمام این عملیات را به طور کامل شرح داده و روش کار هر کدام را همراه با مثال‌های ساده‌ای بیان می‌کنیم.

تعریف دقیق دستورهای CRUD

همان‌طور که متوجه شدیم، برای برقراری ارتباط بین کاربر و منابع مختلف در اپلیکیشن‌ها چهار عملیات اصلی انجام می‌گیرند. این عملیات‌ها Create و Read و Update و Delete هستند. اغلب اوقات، دانشمندان علوم کامپیوتر این دستورها را به شکل خلاصه شده و با کلمه CRUD نام می‌برند. هر مدلی برای کامل بودن باید موفق به انجام این ۴ عملکرد شود. اگر عملیاتی وجود داشت که قابل اجرا توسط هیچ‌کدام از این ۴ دستور نبود، به احتمال زیاد باید به صورت مدل مجزایی تعریف شود.

ایده استفاده از CRUD اغلب در زمان ساخت **وب اپلیکیشن‌ها** به کار برده می‌شود. زیرا برای ساخت مدل‌های کامل و کاربردی، چهارچوبی مناسبی را برای توسعه‌دهندگان فراهم می‌کند. این مدل به سادگی به یاد سپرده می‌شود. برای نمونه، سیستمی را در نظر بگیرید که برای مدیریت کتاب‌های کتابخانه طراحی شده است. در این کتابخانه فرضی، منبع کتابی برای ذخیره‌سازی کتاب وجود دارد. اشیاء این پایگاه داده - کتاب‌ها - را همانند زیر در نظر بگیرید.

```
1 "book": {  
2   "id": <Integer>,  
3   "title": <String>,  
4   "author": <String>,  
5   "isbn": <Integer>  
6 }
```

برای اینکه سیستم کتابداری این کتابخانه قابل استفاده شود، باید مکانیزم واضحی وجود داشته باشد که تمام عملیات CRUD را به طور کامل و صحیح انجام دهد.

Create

این عملیات باید شامل تابعی برای افزودن کتاب به کاتالوگ کتابخانه شود. برنامه‌ای که این تابع را فراخوانی می‌کند، باید اطلاعاتی مانند نام کتاب «`title`»، نام نویسنده «`author`» و شماره شابک «`ISBN`» را به تابع ارسال کند. بعد از فراخوانده شدن این تابع باید شیء کتاب جدیدی به منبع کتاب‌ها `books` افزوده شود. این موجودیت جدید، شامل اطلاعات مربوط به کتاب جدید است. علاوه بر این، شماره شناسه `id` منحصر بفردی هم به آن اختصاص داده می‌شود. با کمک این شماره `id` بعداً به سادگی می‌توانیم به اطلاعات مربوط به کتاب جدید دسترسی داشته باشیم.



Read

این عملکرد شامل تابعی برای فراخوانی کتاب‌هاست. با کمک تابع مورد نظر، تمام کتاب‌های موجود در کاتالوگ قابل دسترسی هستند. در نتیجه اجرای این تابع، هیچ تغییری در کاتالوگ ایجاد نمی‌شود. تابع فراخوانی اطلاعات، فقط داده‌ها را از منبع فراخوانی کرده و نتایج را به کاربر نمایش می‌دهد. عملکرد دیگر این تابع، واکنشی اطلاعات مربوط به موجودیت‌های مختلف به صورت مجزا است. برای جست‌وجو به دنبال هر موجودیت می‌توانیم از تمام اطلاعات مربوط به آن استفاده کنیم. برای مثال، به منظور پیدا کردن کتاب، می‌توان از نام کتاب، نام نویسنده یا شماره `ISBN` به صورت مجزا یا در ترکیب با هم استفاده کرد. این تابع هم هیچ تغییری در اطلاعات فراخوانی شده ایجاد نمی‌کند. فقط اطلاعات را فراخوانی کرده و به کاربر نمایش می‌دهد.

Update

وقتی که اطلاعات مربوط به کتاب خاصی نیاز به تغییر داشته باشند از عملیات Update استفاده می‌کنیم. تابعی که در این شرایط توسط برنامه فراخوانده می‌شود، دارای توانایی اعمال تغییرات در اطلاعات مربوط به موجودیت‌های پایگاه داده است. برای مثال در این زمینه می‌تواند نام کتاب « title »، نام نویسنده « author » و شماره شابک « ISBN » را تغییر دهد. بعد از فراخوانی این تابع بر روی هر کدام از عناصر موجود در کتابخانه، جزئیات مربوط به کتاب مورد نظر با داده‌های جدیدی به‌روزرسانی خواهند شد.

Delete

علاوه بر موارد بالا، تابعی هم باید وجود داشته باشد که بعد از فراخوانده شدن، کتاب‌های مشخص شده را از کاتالوگ حذف کند. برنامه‌ای که این تابع را فراخوانی می‌کند، باید یک یا چند مورد از جزئیات را هم برای آن فراهم کند. در این مثال، تابع با کمک جزئیاتی مانند نام کتاب « title »، نام نویسنده « author » و شماره شابک « ISBN » می‌تواند کتاب مورد نظر را شناسایی و پیدا کند. در نتیجه آن کتاب با موفقیت از منبع کتاب‌ها حذف می‌شود. بعد از اینکه این تابع فراخوانی و اجرا شد، فقط کتاب معرفی شده به این تابع از مخزن حذف می‌شود. باقی کتاب‌ها در سر جای خود بدون تغییر باقی می‌مانند.



آموزش پروژه‌محور برنامه‌نویسی با مثال‌های کاربردی - فرادرس

بهترین فرصت یادگیری؛ هر آموزش فقط ۱۱۹ هزار تومان 📖

آموزش برنامه‌نویسی از پایه با زبان‌های کاربردی - فرادرس

چگونه با کمک فرادرس طراحی سایت بیاموزیم؟

طراحی و برنامه نویسی سایت‌ها تقریباً مهم‌ترین حوزه‌هایی هستند که در آن‌ها متوجه می‌شویم که ضرورت استفاده از CRUD چیست. زیرا دستورهای CRUD بیشتر از همه برای کار با API-ها استفاده می‌شوند. این حوزه یکی از فعال‌ترین حوزه‌های علوم کامپیوتری است. تقریباً تمامی صنایع و کسب‌وکارها برای توسعه و رشد خود به اینترنت متکی هستند. به همین دلیل فضای اشتغال برای

مجموعه آموزش طراحی و برنامه نویسی سایت – مقدماتی تا پیشرفته







آموزش برنامه نویسی پی ایچ پی PHP



یادگیری را شروع کنید <

آموزش اچ تی ام ال – طراحی وب با HTML – مقدماتی



یادگیری را شروع کنید <

آموزش جاوا اسکریپت JavaScript – جامع و کاربردی



یادگیری را شروع کنید <

فردارس هم به عنوان یکی از بزرگترین دانشگاه‌های آنلاین کشور، توجه ویژه‌ای بر روی این حوزه دارد. در مجموعه آموزش طراحی و برنامه نویسی سایت، فیلم‌های آموزشی زیاد و متنوعی تولید و منتشر شده است. با توجه به تنوع تکنولوژی‌هایی که برای فرانت‌اند و بک‌اند سایت‌ها و وب‌اپلیکیشن‌ها به کار برده می‌شوند، لازم است که آموزش‌های متنوع و باکیفیتی نیز به علاقه‌مندان این حوزه ارائه شود. در پایین چند مورد از این فیلم‌های آموزشی را معرفی کرده‌ایم.

- فیلم آموزش طراحی وب با HTML دوره مقدماتی در فرادرس
- فیلم آموزش رایگان آشنایی با برنامه نویسی Back End و Front End و تفاوت آن ها با هم در فرادرس
- فیلم آموزش سئو یا بهینه سازی سایت، اصول SEO در فرادرس

- فیلم آموزش پروژه محور وردپرس WordPress درباره طراحی و پیاده سازی سایت فروش فایل در فرادرس

در صورت تمایل به مشاهده فیلم‌های بیشتر بر روی تصویر بالا کلیک کرده و وارد صفحه اصلی مجموعه فیلم‌های آموزشی مربوط به طراحی و برنامه نویسی سایت‌ها شوید.

ارتباط بین CRUD و REST چیست؟

بیشتر اوقات دستورهای CRUD در محیط REST به ترتیب به متدهای POST و GET و PUT و DELETE - کدهای HTTP - اشاره می‌کنند. این متدها، عناصر بنیادین سیستمی هستند که برای ذخیره‌سازی دائمی اطلاعات به کار برده می‌شود.

در ادامه مطلب، به بیان استانداردها و کدهای پاسخی می‌پردازیم که معمولاً در زمان ساخت اپلیکیشن‌های RESTful توسط توسعه‌دهندگان مورد توجه قرار می‌گیرند. اصول مربوط به استفاده از این متدها ممکن است که در سیستم‌های مختلف فرق کنند. بنابراین، می‌توانید از مقادیر و کدهای مختلفی استفاده کنید.

REST API، سبک معماری برای طراحی سیستم‌های نرم‌افزاری، به ویژه سیستم‌های مبتنی بر وب است. به بیان ساده‌تر، REST API مجموعه‌ای از قوانین و استانداردها را برای تعامل بین نرم‌افزارهای مختلف، تعریف می‌کند. در تمام زبان‌های برنامه نویسی که برای طراحی سایت و نوشتن بک‌اند به کار برده می‌شوند، این سبک معماری به رسمیت شناخته می‌شود. برای مثال، به منظور آشنایی با آن در وردپرس می‌توانید فیلم آموزش REST API در وردپرس برای کار با داده‌های پایگاه داده را از فرادرس مشاهده کنید. لینک مربوط به این فیلم در پایین نیز قرار داده شده است.

فیلم آموزش REST API در وردپرس در فرادرس

کلیک کنید



برای ادامه مطلب، فرض کنید که در حال کار بر روی سیستمی برای نظارت بر غذاهای رستوران و قیمت هر کدام هستیم. در ادامه می‌بینیم که در چنین سیستمی روش پیاده‌سازی عملیات مربوط به CRUD چیست.

دستور Create در REST

در محیط REST، رایج‌ترین روش برای ساخت منبع داده استفاده از متد `POST` در HTTP است. دستور `POST` برای ساخت منبع داده جدید و افزودن عناصر مرتبط به آن منبع داده استفاده می‌شود.

برای مثال، فرض کنیم که می‌خواهیم غذای جدیدی را به فهرست غذای‌های روزانه رستوران اضافه کنیم. شیء `dish` در منبع `dishes` ذخیره شده است. اگر بخواهیم که آیتم جدیدی ایجاد کنیم باید درخواست `POST` را به کار ببریم.

در خواست مورد نظر به شکل زیر نوشته می‌شود.

```
1 POST http://www.myrestaurant.com/dishes/
2 Body -
3
4 {
5   "dish": {
6     "name": "Avocado Toast",
7     "price": 8
8   }
9 }
```

کدهای بالا، آیتم جدیدی را با مقدار `Avocado Toast` به ازای نام `name` و مقدار `8` به ازای قیمت `price` ایجاد می‌کند. در صورتی که عملیات ساخت عنصر جدید در پایگاه داده با موفقیت انجام شود، سرور پاسخی را برمی‌گرداند که شامل `Header` و بدنه `Response` است. داخل بخش `Header` در پاسخ، لینکی به منبع ساخته شده جدید وجود دارد و در بخش `Body` پاسخ کد `201` قرار داده می‌شود. این کد به معنای پایان موفقیت‌آمیز عملیات ساخت عنصر در منبع است. پاسخ برگشت داده شده از سرور به شکل زیر است.



آموزش پروژه محور Python با
تمرین‌های عملی - فرادرس



آموزش کدنویسی راحت‌تر از
همیشه با ChatGPT - فرادرس



آموزش ساخت اپلیکیشن با
PySide در Python - فرادرس

```
1 Status Code - 201 (CREATED)
2
3 Body -
4
5 {
6   "dish": {
7     "id": 1223,
8     "name": "Avocado Toast",
9     "price": 8
10  }
11 }
```

با توجه به پاسخ برگشت داده شده از سرور، می‌توان مشاهده کرد که اطلاعات مربوط به غذای Avocado Toast با قیمت 8 با موفقیت ساخته شده و به منبع لیست غذاها نیز اضافه شد.



دستور Read در REST

دستورات CRUD جزء بسیار مهمی از عملیات مربوط به REST API هستند که در زبان‌های مختلف برنامه نویسی به کار برده می‌شوند. برای مثال زبان پایتون به عنوان یکی از زبان‌های قدرتمند برای نوشتن بک‌اند شناخته می‌شود. در صورتی که بخواهید روش نوشتن دستورات مربوط به **REST API** **در پایتون** را بیاموزید، پیشنهاد می‌کنیم که مطلب مربوط به آن را در مجله فرادرس مطالعه کنید.

برای خواندن منابع در محیط REST، از متد **GET** استفاده می‌کنیم. عملیات خواندن اطلاعات، هرگز نباید هیچ تغییری در اطلاعات موجود در منبع ایجاد کند. فقط باید اطلاعات مورد نظر را فراخوانی کند. اگر متد **GET** را ده بار پشت سر هم بر روی اطلاعات یکسانی فراخوانی کنیم، بار دهم هم باید همان پاسخی را دریافت کنیم که در بار اول دریافت کرده بودیم. با کمک متد **GET** حتی می‌توانیم کل آیتم‌های ذخیره شده در لیست را فراخوانی کرده و مشاهده کنیم.

```
1 GET http://www.myrestaurant.com/dishes/  
2 Response: Status Code - 200 (OK)  
3  
4 Body -  
5  
6 {
```

```

7   "dishes": [
8     {
9       "id": 1,
10      "name": "Spring Rolls",
11      "price": 6
12    },
13    {
14      "id": 2,
15      "name": "Mozzarella Sticks"

```

مشاهده کامل کدها

استفاده از متد **GET** برای ارسال درخواست‌های HTTP می‌توان اطلاعات عنصرها را به صورت نزا از دیگران هم مطالعه کرد. برای انجام این کار لازم است که **id** مربوط به آن عنصر را به صورت ریح در درخواست مشخص کنیم.

```

1  GET http://www.myrestaurant.com/dishes/1223
2  Response: Status Code - 200 (OK)
3
4  Body -
5
6  {
7    "id": 1223,
8    "name": "Avocado Toast",
9    "price": 8
10 }

```

بعد از ارسال این درخواست هیچ اطلاعاتی در پایگاه داده تغییر نمی‌کند. فقط اطلاعات مربوط به آیتم شماره 1223 از منبع غذاها - بدون هیچ نوع تغییری - فراخوانی می‌شوند. اگر در زمان ارسال دستور و پردازش آن با خطا روبه‌رو نشویم، پاسخ درخواست ارسال شده به صورت HTML یا **JSON** برگردانده می‌شود. همراه با پاسخ، کد وضعیت 200 - به معنای «OK» - برگردانده می‌شود. اما اگر مشکلی در درخواست وجود داشته باشد، اغلب اوقات کد 404 به کامپیوتر صادر کننده درخواست برگردانده می‌شود. کد 404 به معنای «پیدا نشده» (NOT FOUND) است.

دستور Update در REST

به منظور اجرای عملیات Update - سومین عملیات در CRUD - باید از متد PUT برای ارسال درخواست HTTP استفاده کنیم.

به عنوان مثال، اگر قیمت غذای Avocado Toast بالا رفته باشد، باید به سراغ پایگاه داده رفته و اطلاعات مربوط به قیمت این ردیف را به روزرسانی کنیم. برای انجام این کار، با استفاده از متد PUT درخواست خود را ارسال می‌کنیم.

```
1 PUT http://www.myrestaurant.com/dishes/1223
2 Body -
3
4 {
5   "dish": {
6     "name": "Avocado Toast",
7     "price": 10
8   }
9 }
```

درخواستی که در کادر بالا ارسال شده، باید آیتمی را که id با شماره 1223 دارد پیدا کرده و اطلاعات فراهم شده در بدنه درخواست را در این آیتم تغییر دهد. با توجه به درخواست بالا، غذا با شماره id مشخص شده - 1223 - از پایگاه داده نام قبلی خود را که Avocado Toast بود نگهداشته است. اما قیمت غذا بیشتر شده و برابر با 10 واحد پولی می‌شود. این قیمت قبلاً برابر با 8 واحد بود.



بعد از ارسال درخواست بالا و در صورتی که سرور برای اجرای آن به مشکلی برخورد، پیغام زیر به عنوان نتیجه اجرای دستور به کاربر برگردانده می‌شود.

```
1 Response: Status Code - 200 (OK)
2
3 Body -
4
5 {
6   "dish": {
7     "name": "Avocado Toast",
8     "price": 10
9   }
10 }
```

پاسخ نمایش داده شده در بالا حاوی کد وضعیت 200 است. این کد به معنای موفقیت در انجام درخواست ارسال شده به سرور است. البته بعضی وقت‌ها هم کد 204 به معنای «بدون محتوا» (NO CONTENT) به کاربر برگردانده می‌شود. این کد هم به معنای موفقیت در اجرای درخواست است اما هیچ داده‌ای را در بدنه پاسخ برنمی‌گرداند. کد 204 بسته به موقعیت شاید به کاربر برگردد.

دستور Delete در REST

دستور Delete از عملیات CRUD توسط متد DELETE در HTTP اجرا می‌شود. این دستور برای حذف کردن منابع مشخص شده در سرور به کار برده می‌شود.

برای مثال فرض کنیم که به دلیل کمبود آوکادو در رستوران دیگر قادر به ارائه غذای Avocado Toast نیستیم. اکنون باید به سراغ پایگاه داده رفته و ردیف مربوط به غذای Avocado Toast را از آن حذف کنیم. با توجه به مطالب بالا می‌دانیم که این غذا با id به شماره 1223 در پایگاه داده ذخیره شده است.

برای حذف این غذا از پایگاه داده باید درخواست خود را به شکل زیر ارسال کنیم.

```
1 DELETE http://www.myrestaurant.com/dishes/1223
```

برای چنین درخواستی، اگر دستور با موفقیت اجرا شود، سرور پاسخی را با کد وضعیت 204 به معنای «بدون محتوا» (NO CONTENT) برمی‌گرداند. در این کد وضعیت هیچ بدنه پاسخی وجود ندارد. بعد از اجرای این درخواست، دیگری هیچ غذایی dish با شماره شناسایی 1223 در منبع غذاها dishes وجود ندارد. پاسخ این دستور به شکل Status Code - 204 (NO CONTENT) است.

بعد از اجرا شدن دستور DELETE بر روی پایگاه داده رستوران، اگر دستور GET را بر روی منبع dishes فراخوانی کنیم، فهرست کامل غذاهای موجود از پایگاه داده برگردانده می‌شود. اما در این فهرست دیگر ردیفی با اطلاعات {id: 1223, "name": "Avocado Toast", "price": 10} وجود ندارد. زیرا این ردیف داده اکنون حذف شده است. بقیه اشیاء dish در پایگاه داده dishes باید بدون تغییر باقی بمانند. در این صورت اگر دستور GET را همراه با id به شماره 1223 فراخوانی کنیم، سرور پاسخی با کد وضعیت 404 برگردانده و هیچ کار دیگری انجام نمی‌دهد. زیرا این شماره id از پایگاه داده حذف شده است. کد وضعیت 404 به معنای «پیدا نشده» (NOT FOUND) است.



📺 فرصت طلایی: اشتراک فرادرس با ۷۵٪ تخفیف

➤ اتصال Python به پایگاه داده و کار با SQL - فرادرس

➤ مثال‌های کاربردی برای یادگیری سریع Python - فرادرس

آشنایی و تسلط بر روی دیتابیس‌های مختلف

بعد از اینکه متوجه شدیم CRUD چیست، در بخش بعدی مطلب به بیان دستورات معادل CRUD در پایگاه‌های داده رابطه‌ای پرداخته‌ایم. هر چند در صورت تمایل به کار کردن بر روی پایگاه‌های داده لازم است که با دستورات بیشتری نیز آشنا شویم. پایگاه‌های داده متنوعی وجود دارند. در بعضی از این پایگاه‌های داده از زبان SQL با سینتکس مشابه هم استفاده می‌شود. بعضی دیگر هم از قواعد NoSQL برای کوئری‌نویسی استفاده می‌کنند. در این بخش، فیلم‌های آموزشی مربوط به چند مورد از این پایگاه‌های داده را معرفی کرده‌ایم. با توجه به پروژه خود تصمیم می‌گیریم که از چه پایگاه داده‌ای استفاده کنیم.

- فیلم آموزش پایگاه داده MySQL در فرادرس
- فیلم آموزش SQL Server دوره تکمیلی فرادرس
- فیلم آموزش رایگان مبانی و مقدمات پایگاه داده کلیک هاوس ClickHouse در فرادرس
- فیلم آموزش پایگاه داده مونگو دی بی MongoDB برای تحلیل، مدل سازی و پردازش داده ها در فرادرس
- فیلم آموزش کلان داده و پایگاه داده غیر رابطه ای NOSQL در فرادرس



با کلیک بر روی تصویر بالا می‌توانید به صفحه اصلی مجموعه فیلم‌های آموزش پایگاه داده هدایت شوید.

در ادامه می‌بینیم که در پایگاه‌های داده رابطه‌ای، دستورات معادل CRUD چیست.

پیاده سازی دستورات CRUD در پایگاه داده

همین‌طور که اشاره شد، دستورات CRUD در اپلیکیشن‌های مربوط به پایگاه داده هم استفاده می‌شوند. از این دستورات چهارگانه به شکل بسیار رایجی در نرم‌افزارهای مربوط به «مدیریت پایگاه‌های داده رابطه‌ای» (Relational Database Management Systems | RDBMS) مانند PostgreSQL و MySQL و Oracle و غیررابطه‌ای NoSQL مانند MongoDB و Apache Cassandra و AWS DynamoDB نیز به کار برده می‌شوند.



از عملیات مشابه با CRUD حتی می‌توانیم بر روی ساختارهای دائمی داده مانند فایل‌ها نیز استفاده کنیم. برای مثال می‌توانیم در مایکروسافت ورد کارهای زیر را انجام دهیم.

- فایل متنی ایجاد کنیم.
- اطلاعات متنی داخل این سند را به‌روزرسانی کنیم.
- هر وقت که خواستیم داده‌های نوشته شده را بخوانیم.
- در صورت نیاز تمام یا بخشی از سند را حذف کنیم.

البته فایل‌ها مانند رکوردهای ذخیره شده در پایگاه‌های داده نیستند. اصطلاح CRUD بیشتر برای عملیات مرتبط با رکوردهای پایگاه‌های داده استفاده می‌شود تا فایل‌هایی مانند فایل‌های متنی در ورد.

در پایگاه داده روش پیاده سازی CRUD چیست؟

در این قسمت از مطلب، نوشتن دستورات CRUD را با استفاده از دستورات «زبان کوئری نویسی ساخت‌یافته» (Structure Query Language | SQL) نمایش داده‌ایم.

به ازای دستور CREATE از عبارت INSERT استفاده می‌شود.

```
1 INSERT INTO <table></table> VALUES (field value 1, field value,
```



به ازای دستور READ از عبارت SELECT استفاده می‌شود.

```
1 SELECT field 1, field 2, ...FROM <table></table> [WHERE ]
```

عبارت UPDATE برای پیاده‌سازی دستور UPDATE به کار برده می‌شود.

```
1 UPDATE <table></table> SET field1=value1, field2=value2,... [WHERE ]
```



عبارت DELETE برای پیاده‌سازی دستور DELETE به کار برده می‌شود.

```
1 DELETE FROM <table></table> [WHERE ]
```

جمع‌بندی

در این مطلب متوجه شدیم که CRUD چیست و از چه دستورهای تشکیل شده است. به طور خلاصه این عبارت ۴ عملیات رایج و ضروری بر روی پایگاه داده را نشان می‌دهد.

۱. **دستور CREATE:** برای مثال کاربران در شبکه‌های اجتماعی باید بتوانند حساب باز کنند و اطلاعات شخصی خود را وارد کنند.
۲. **دستور READ:** هم خود کاربر و هم سایر مردم به منظور جست‌وجو به دنبال کاربر باید امکان دسترسی به اطلاعات وی را داشته باشند.
۳. **دستور UPDATE:** هر کسی با توجه به شرایط باید امکان به‌روزرسانی اطلاعات شخصی خود را داشته باشد.
۴. **دستور DELETE:** هرکسی که از فعالیت در شبکه اجتماعی خسته شده باید امکان حذف اکانت خود را به طور کامل داشته باشد.

فیلم مجموعه آموزش طراحی سایت با برنامه نویسی - مقدماتی تا پیشرفته در فرادرس

کلیک کنید



در این مطلب از مجله فرادرس، با مفهوم CRUD آشنا شدیم. ابتدا این مفهوم را تعریف کرده و گفتیم که منظور از CRUD چیست. سپس دستورات تشکیل دهنده CRUD را معرفی کردیم. بعد از آن تمام این دستورات را همراه با مثال ساده‌ای یک به یک توضیح داده و کدهای مربوط به آن‌ها را نمایش دادیم. در نهایت هم دستورات معادل عملیات CRUD را در زبان کوئری نویسی SQL معرفی کردیم.

محصول، خدمات یا برند خود را در مجله فرادرس معرفی کنید. [کلیک کنید](#)

اگر این مطلب برای شما مفید بوده است، آموزش‌ها و مطالب زیر نیز به شما پیشنهاد می‌شوند:

مجموعه آموزش برنامه نویسی وب - مقدماتی تا پیشرفته

آموزش فریم ورک جنگو در پایتون - ساخت Web APIs با Django Rest

مجموعه آموزش برنامه نویسی پی‌اچ‌پی PHP - مقدماتی تا پیشرفته

Api در جاوا اسکریپت - از صفر تا صد + مثال و کد

کدنویسی REST API با پایتون - از صفر تا صد

CrowdStrike

freeCodeCamp

codecademy

منابع: 

مصطفی رشیدی (+)

مصطفی رشیدی دانش‌آموخته مقطع لیسانس مهندسی نرم افزار است. علاقه بسیار زیادی به هوش مصنوعی و دیتاساینس دارد. برای مجله فرادرس در زمینه هوش مصنوعی و برنامه‌نویسی و مطالب مرتبط با علوم کامپیوتری می‌نویسد.



مطالب مرتبط

کار با رشا

اصول نام گذاری در برنامه نویسی تمیز - به زبان ساده

شروع یادگیری ←

وایب کدینگ چیست و چطور Vibe Coding یاد بگیریم؟

شروع یادگیری ←

شما در حال مطالعه نسخه آفلاین یکی از مطالب «مجله فرادرس» هستید. لطفاً توجه داشته باشید، ممکن است برخی از قابلیت‌های تعاملی مطالب، مانند امکان پاسخ به پرسش‌های چهار گزینه‌ای و مشاهده جواب صحیح آن‌ها، نمایش نتیجه آزمون‌ها، پاسخ تشریحی سوالات، پخش فایل‌های صوتی و تصویری و غیره، در این نسخه در دسترس نباشند. برای دسترسی به نسخه آنلاین مطلب، استفاده از کلیه امکانات آن و داشتن تجربه کاربری بهتر [اینجا کلیک کنید](#).