CORS là gì? Giới thiệu tất tần tật về CORS

By

**[thao le](https://topdev.vn/blog/author/thaole/)**

16473

[Share on Facebook](https://www.facebook.com/sharer.php?u=https%3A%2F%2Ftopdev.vn%2Fblog%2Fcors-la-gi%2F" \t "_new)

[Tweet on Twitter](https://twitter.com/intent/tweet?text=CORS+l%C3%A0+g%C3%AC%3F+Gi%E1%BB%9Bi+thi%E1%BB%87u+t%E1%BA%A5t+t%E1%BA%A7n+t%E1%BA%ADt+v%E1%BB%81+CORS&url=https%3A%2F%2Ftopdev.vn%2Fblog%2Fcors-la-gi%2F&via=TopDev" \t "_new)

* [tweet](https://twitter.com/share)

[](https://topdev.vn/blog/wp-content/uploads/2019/04/cors.png)

Một nhu cầu rất thông dụng với các [developer web](https://topdev.vn/blog/con-duong-tro-thanh-cao-thu-web-developer-ban-chon-huong-di-nao/) đó là truy truy vấn qua [API](https://topdev.vn/blog/restful-api-la-gi/). Tuy nhiên, việc truy vấn và xử lý dữ liệu từ API nhiều khi cũng rất khó khăn.

Rất nhiều lập trình viên phải đối mặt với các vấn đề liên quan đến CORS. Vậy CORS là gì? Và lý do tại sao chúng ta cần CORS?

**CORS là gì ?**

**CORS** là một cơ chế cho phép nhiều tài nguyên khác nhau (fonts, Javascript, v.v…) của một trang web có thể được truy vấn từ domain khác với domain của trang đó. **CORS** là viết tắt của từ Cross-origin resource sharing.

[Những việc làm hấp dẫn](https://topdev.vn/)

* [](https://topdev.vn/partners/detail-jobs/mobile-apps-developer-4-15093/?utm_source=topdev.vn&utm_medium=jrec&utm_term=Mobile+Apps+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==)

**[Mobile Apps Developer](https://topdev.vn/partners/detail-jobs/mobile-apps-developer-4-15093/?utm_source=topdev.vn&utm_medium=jrec&utm_term=Mobile+Apps+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "Mobile Apps Developer" \t "_blank)**

**[Kyanon Digital](https://topdev.vn/partners/detail-jobs/mobile-apps-developer-4-15093/?utm_source=topdev.vn&utm_medium=jrec&utm_term=Mobile+Apps+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "Mobile Apps Developer" \t "_blank)***[HCMNegotiable](https://topdev.vn/partners/detail-jobs/mobile-apps-developer-4-15093/?utm_source=topdev.vn&utm_medium=jrec&utm_term=Mobile+Apps+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "Mobile Apps Developer" \t "_blank)*

[MobileReact NativeApp](https://topdev.vn/partners/detail-jobs/mobile-apps-developer-4-15093/?utm_source=topdev.vn&utm_medium=jrec&utm_term=Mobile+Apps+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "Mobile Apps Developer" \t "_blank)

* [](https://topdev.vn/partners/detail-jobs/c-developer-machine-leadningai-2010635/?utm_source=topdev.vn&utm_medium=jrec&utm_term=C%2B%2B+Developer+%28Machine+Leadning%2FAI%29&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==)

**[C++ Developer (Machine Leadning/AI)](https://topdev.vn/partners/detail-jobs/c-developer-machine-leadningai-2010635/?utm_source=topdev.vn&utm_medium=jrec&utm_term=C%2B%2B+Developer+%28Machine+Leadning%2FAI%29&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "C++ Developer (Machine Leadning/AI)" \t "_blank)**

**[Koh Young Việt Nam](https://topdev.vn/partners/detail-jobs/c-developer-machine-leadningai-2010635/?utm_source=topdev.vn&utm_medium=jrec&utm_term=C%2B%2B+Developer+%28Machine+Leadning%2FAI%29&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "C++ Developer (Machine Leadning/AI)" \t "_blank)***[HNNegotiable](https://topdev.vn/partners/detail-jobs/c-developer-machine-leadningai-2010635/?utm_source=topdev.vn&utm_medium=jrec&utm_term=C%2B%2B+Developer+%28Machine+Leadning%2FAI%29&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "C++ Developer (Machine Leadning/AI)" \t "_blank)*

[C++AIMachine Learning](https://topdev.vn/partners/detail-jobs/c-developer-machine-leadningai-2010635/?utm_source=topdev.vn&utm_medium=jrec&utm_term=C%2B%2B+Developer+%28Machine+Leadning%2FAI%29&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "C++ Developer (Machine Leadning/AI)" \t "_blank)

* [](https://topdev.vn/partners/detail-jobs/remote-full-stack-developer-2-2010538/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BRemote%5D+Full-stack+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==)

**[[Remote] Full-stack Developer](https://topdev.vn/partners/detail-jobs/remote-full-stack-developer-2-2010538/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BRemote%5D+Full-stack+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "[Remote] Full-stack Developer" \t "_blank)**

**[IMIAI INC](https://topdev.vn/partners/detail-jobs/remote-full-stack-developer-2-2010538/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BRemote%5D+Full-stack+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "[Remote] Full-stack Developer" \t "_blank)***[HCMNegotiable](https://topdev.vn/partners/detail-jobs/remote-full-stack-developer-2-2010538/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BRemote%5D+Full-stack+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "[Remote] Full-stack Developer" \t "_blank)*

[NodeJSFull-StackReact](https://topdev.vn/partners/detail-jobs/remote-full-stack-developer-2-2010538/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BRemote%5D+Full-stack+Developer&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "[Remote] Full-stack Developer" \t "_blank)

* [](https://topdev.vn/partners/detail-jobs/back-end-developer-80-14779/?utm_source=topdev.vn&utm_medium=jrec&utm_term=BACK-END+DEVELOPER&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==)

**[BACK-END DEVELOPER](https://topdev.vn/partners/detail-jobs/back-end-developer-80-14779/?utm_source=topdev.vn&utm_medium=jrec&utm_term=BACK-END+DEVELOPER&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "BACK-END DEVELOPER" \t "_blank)**

**[FIIN](https://topdev.vn/partners/detail-jobs/back-end-developer-80-14779/?utm_source=topdev.vn&utm_medium=jrec&utm_term=BACK-END+DEVELOPER&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "BACK-END DEVELOPER" \t "_blank)***[HN12,000,000 - 25,000,000 VND](https://topdev.vn/partners/detail-jobs/back-end-developer-80-14779/?utm_source=topdev.vn&utm_medium=jrec&utm_term=BACK-END+DEVELOPER&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "BACK-END DEVELOPER" \t "_blank)*

[Back-EndNodeJSPHP](https://topdev.vn/partners/detail-jobs/back-end-developer-80-14779/?utm_source=topdev.vn&utm_medium=jrec&utm_term=BACK-END+DEVELOPER&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "BACK-END DEVELOPER" \t "_blank)

* [](https://topdev.vn/partners/detail-jobs/all-level-lap-trinh-vien-java-2010598/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BAll+Level%5D+L%E1%BA%ACP+TR%C3%8CNH+VI%C3%8AN+JAVA&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==)

**[[All Level] LẬP TRÌNH VIÊN JAVA](https://topdev.vn/partners/detail-jobs/all-level-lap-trinh-vien-java-2010598/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BAll+Level%5D+L%E1%BA%ACP+TR%C3%8CNH+VI%C3%8AN+JAVA&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "[All Level] LẬP TRÌNH VIÊN JAVA" \t "_blank)**

**[Citynow](https://topdev.vn/partners/detail-jobs/all-level-lap-trinh-vien-java-2010598/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BAll+Level%5D+L%E1%BA%ACP+TR%C3%8CNH+VI%C3%8AN+JAVA&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "[All Level] LẬP TRÌNH VIÊN JAVA" \t "_blank)***[HCMNegotiable](https://topdev.vn/partners/detail-jobs/all-level-lap-trinh-vien-java-2010598/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BAll+Level%5D+L%E1%BA%ACP+TR%C3%8CNH+VI%C3%8AN+JAVA&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "[All Level] LẬP TRÌNH VIÊN JAVA" \t "_blank)*

[Restful ApiCSS3ReactJSHTML5Java](https://topdev.vn/partners/detail-jobs/all-level-lap-trinh-vien-java-2010598/?utm_source=topdev.vn&utm_medium=jrec&utm_term=%5BAll+Level%5D+L%E1%BA%ACP+TR%C3%8CNH+VI%C3%8AN+JAVA&token=RIBPOSVJgL+kR9V78tC1PePVkAR8STLmwYHXhY4o2z+8Jg6rR09xCNe2Zk2K/rcY1dv3kOtiRmtj5J4rtFRYiw==" \o "[All Level] LẬP TRÌNH VIÊN JAVA" \t "_blank)

**Tại sao chúng ta cần CORS**

CORS được sinh ra là vì [same-origin policy](https://www.w3.org/Security/wiki/Same_Origin_Policy), là một chính sách liên quan đến bảo mật được cài đặt vào toàn bộ các trình duyệt hiện nay. Chính sách này ngăn chặn việc truy cập tài nguyên của các domain khác một cách vô tội vạ.

Ta có ví dụ một kịch bản như sau:

* Bạn truy cập một trang web có mã độc. Trang web đó sử dụng Javascript để truy cập tin nhắn Facebook của bạn ở địa chỉ https://facebook.com/messages.
* Nếu bạn đã đăng nhập Facebook từ trước rồi. Nếu không có same-origin policy, trang web độc hại kia có thể thoải mái lấy dữ liệu của bạn và bất cứ điều gì chúng muốn.

Same-origin policy chính là để ngăn chặn những kịch bản như trên để bảo vệ người dùng, giúp an toàn hơn khi lướt web. Bạn có thể thử trên web console và sẽ nhận được lỗi ngay:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | $.get('https://facebook.com/messages')  Access to XMLHttpRequest at 'https://facebook.com/messages' from  origin 'xxx' has been blocked by CORS policy: No  'Access-Control-Allow-Origin' header is present on the requested  resource. |

Truy cập URL trên từ bất kỳ domain nào ngoài facebook.com bạn cũng sẽ nhận được lỗi như vậy. Đó chính là nhờ same-origin policy.

Thế nhưng trong thế giới web, lập trình viên thường xuyên phải thực hiện truy vấn đến các domain khác, đặc biệt là khi làm việc với các API.

Đó là lúc chúng ta cần đến **CORS**. **CORS** sử dụng các HTTP header để “thông báo” cho trình duyệt rằng, một ứng dụng web chạy ở origin này (thường là domain này) có thể truy cập được các tài nguyên ở origin khác (domain khác).

Một ứng dụng web sẽ thực hiện truy vấn HTTP cross-origin nếu nó yêu cầu đến các tài nguyên ở origin khác với origin nó đang chạy (khác giao thức, domain, port). Sự khác biệt về giao thức ở đây là khác biệt kiểu như HTTP với FTP chứ không phải HTTP và HTTPS (dù nhiều trình duyệt không cho phép trộn lẫn các tài nguyên truy cập bằng HTTP và HTTPS nhưng đó là vấn đề khác, không liên quan đến CORS).

Các trường hợp cần đến CORS rất phổ biến trong thực tế. Một ví dụ rất điển hình như sau: một ứng dụng web chạy ở domain foo.com và nó cần truy vấn đến bar.com để lấy một vài dữ liệu (thường được thực hiện bởi JavaScript bằng cách sử dụng [XMLHttpRequest](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest" \t "_new)).

Các trình duyệt đều cài đặt same-origin policy và tuân thủ nó rất chặt chẽ. Cài đặt XMLHttpRequest và kể cả [Fetch API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API) cũng đều tuân thủ chính sách này. Do đó những truy vấn như ở trên sẽ không thu được kết quả gì, trừ khi máy chủ trả về response có các header CORS phù hợp.

Như vậy, bằng việc sử dụng CORS, chúng ta có thể thúc đẩy việc giao tiếp trong ứng dụng web dễ dàng hơn rất nhiều.

**Các truy vấn dùng CORS**

Các truy vấn sau bắt buộc phải sử dụng CORS, theo tiêu chuẩn [quốc tế](https://fetch.spec.whatwg.org/" \l "http-cors-protocol" \t "_new).

* Các truy vấn bằng XMLHttpRequest hoặc Fetch API đến một domain khác.
* [WebGL Texture](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Using_textures_in_WebGL)
* Ảnh, video được vẽ vào canvas sử dụng [drawImage](https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/drawImage" \t "_new).
* Web fonts truy vấn đến domain khác qua @fontface của CSS, trong đó trang web chỉ có thể sử dụng font dạng True Type nếu được cho phép.

**[RESTful API là gì? Cách thiết kế RESTful API](https://topdev.vn/blog/restful-api-la-gi/" \t "_blank)**

**[Những API thần thánh mà không phải ai cũng biết](https://topdev.vn/blog/nhung-api-than-thanh-ma-khong-phai-ai-cung-biet/" \t "_blank)**

**Làm thế nào để sử dụng CORS**

Một hiểu lầm khá phổ biến, nhất là với các lập trình viên mới làm việc với API lại được làm việc với API của các hãng lớn, tài liệu đầy đủ, đó là cho rằng CORS là công việc của frontend. Nhưng thực ra CORS hoàn toàn là [công việc của backend](https://topdev.vn/blog/lo-trinh-tro-thanh-backend-developer/).

Các lập trình viên frontend thường không cần phải thao tác nhiều nếu cần dùng đến các truy vấn CORS (trừ một số ngoại lệ như không được sử dụng thư viện hoặc phải hỗ trợ IE 8). Khi một trình duyệt gửi một truy vấn đến máy chủ, nó sẽ tự động thiết lập một số HTTP header (ví dụ Origin) chứa các thông tin về nguồn gốc của truy vấn đó.

Về phía máy chủ, sau khi có được thông tin về nguồn gốc của truy vấn, nó có thể lựa chọn không phải hồi truy vấn đó, trả về lỗi hoặc trả về dữ liệu cần thiết. Trong trường hợp trả về dữ liệu, máy chủ cần thiết lập các HTTP header sao cho trình duyệt hiểu rằng truy vấn đó đã được chấp nhận.

Như vậy,chúng ta có thể thấy rằng, CORS giúp thúc đấy quá trình trao đổi dữ liệu giữa trình duyệt và máy chủ. CORS hoàn toàn không có liên quan gì đến việc trao đổi trực tiếp giữa ứng dụng web mà một máy chủ web khác, ví dụ backend của ứng dụng đó truy cập đến tài nguyên trên một origin khác, nó cũng không cần đến CORS.

**Tạo truy vấn CORS bằng XMLHttpRequest**

Trong phần này chúng ta sẽ tìm hiểu cách tạo ra các truy vấn CORS bằng JavaScript. CORS được hỗ trợ bởi hầu hết các trình duyệt hiện đại. Riêng với IE, nó chỉ hỗ trợ từ IE 8 trở lên mà thôi.

Tạo truy vấn

Các trình duyệt Chrome, Firefox, Safari đều sử dụng version mới của XMLHttpRequest do đó việc truy vấn CORS diễn ra hết sức thuận lợi. IE thì sử dụng XDomainRequest, nó hoạt động gần giống với XMLHttpRequest nhưng có nhiều hạn chế hơn.

Chúng ta có thể bắt đầu bằng cách tạo ra các object cần thiết. Dưới đây là một đoạn code như thế:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | const createCORSRequest = (method, url) => {      let xhr = new XMLHttpRequest();      if ('withCredentials' in xhr) {          // Kiểm tra XMLHttpRequest object có thuộc tính  // withCredentials hay không          // Thuộc tính này chỉ có ở XMLHttpRequest2          xhr.open(method, url, true);      } else if (typeof XDomainRequest != 'undefined') {          // Kiểm tra XDomainRequest          // Đây là đối tượng chỉ có ở IE và  // là cách để IE thực hiện truy vấn CORS          xhr = new XDomainRequest();          xhr.open(method, url);      } else {          xhr = null;      }      return xhr;  }    const request = createCORSRequest('GET',  'https://jsonplaceholder.typicode.com/posts/1');  if (!request) {      throw new Error('CORS is not supported');  } |

Sau khi tạo được đối tượng XMLHttpRequest rồi thì chúng ta cần một số event handler, trong trường hợp này, chúng ta chỉ cần quan tâm 2 event onload và onerror là đủ. Ngoài ra còn một số event khác như ontimeout, onprogress không được sử dụng nhiều lắm.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | request.onload = () => {      const responseText = request.responseText;      console.log(responseText);  }  request.onerror = () => {      console.log('Error');  } |

Thực ra các trình duyệt khác nhau lại có cách cài đặt rất khác nhau với event onerror. Ví dụ, Firefox trả về status là 0 và statusText luôn rỗng với mọi lỗi. Ngoài ra, các trình duyệt cũng thường không cho phép truy cập đến nội dung cụ thể của lỗi đã xảy ra, chúng ta chỉ biết rằng đã có lỗi mà thôi.

withCredentials

Mặc định, các truy vấn CORS không gửi hoặc thiết lập bất cứ cookie nào trên trình duyệt. Nếu muốn sử dụng cookie trong truy vấn đó, chúng ta phải đặt thuộc tính withCredentials của truy vấn bằng true:

|  |  |
| --- | --- |
| 1  2  3 | xhr.withCredentials = true; |

Tuy nhiên, đó cũng mới chỉ là một nửa mà thôi. Nửa còn lại thuộc về phía máy chủ, đó là HTTP header Access-Control-Allow-Credentials phải là true (chúng ta sẽ tìm hiểu ở phần sau).

Với giá trị withCredentials bằng true, cookie sẽ được tự động thêm vào cũng như thiết lập nếu có phản hồi từ máy chủ. Lưu ý rằng, cookie trong trường hợp này là third-party cookie và việc lưu trữ, truy cập cookie vẫn hoàn toàn thuân theo same-origin policy, do đó, chúng ta không thể truy cập cookie bằng document.cookie được. Nó hoàn toàn được xử lý tự động bởi trình duyệt.

Gửi truy vấn

Sau khi mọi việc đã hoàn tất, việc cuối cùng chúng ta cần làm là gửi truy vấn đi nữa mà thôi:

|  |  |
| --- | --- |
| 1  2  3 | request.send(); |

Lúc này truy vấn sẽ được gửi đến máy chủ, và nếu máy chủ đó chấp nhận CORS thì nó sẽ trả về response tương ứng. Hoạt động của truy vấn lúc này hoàn toàn giống với truy vấn có chúng origin thông thường.

**Tạo truy vấn CORS bằng jQuery**

Hàm $.ajax của jQuery có thể được sử dụng cho các truy vấn thông thường lẫn truy vấn CORS (cookie cũng được hỗ trợ mặc định). Do đó nếu sử dụng jQuery thì công việc của lập trình viên cũng khá dễ dàng. Tuy nhiên, cần lưu ý một số điều như sau:

* Truy vấn CORS của jQuery không hỗ trợ object XDomainRequest của IE, chúng ta cần sử dụng thêm plugin để hỗ trợ việc này.
* Giá trị $.support.cors sẽ được gán là true nếu trình duyệt hỗ trợ CORS (với IE sẽ là false). Giá trị này có thể được sử dụng để kiểm tra xem CORS có được hỗ trợ hay không.

Dưới đây là một đoạn code sử dụng jQuery để tạo truy vấn CORS:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | $.ajax({  type: 'GET',  url: 'https://jsonplaceholder.typicode.com/posts/1',  success: data => {  console.log(data);  },  error: () => {  console.log('Error');  }  }) |

**Tạo truy vấn CORS bằng Fetch API**

Chúng ta cũng có thể sử dụng Fetch API để tạo truy vấn CORS. Tuy nhiên, fetch mới chỉ xuất hiện từ [ES6](https://topdev.vn/blog/meo-voi-javascript-es6-va-thu-thuat-de-lam-cho-code-sach-hon-ngan-hon-va-de-doc-hon-phan-1-2/) nên nhiều trình duyệt vẫn chưa hỗ trợ nó (cụ thể là IE tất cả các phiên bản đều không hỗ trợ).

Fetch API cho chúng ta một phương thức đơn giản để tạo các truy vấn, và nó đã cài đặt sẵn việc hỗ trợ CORS nên chúng ta cũng có thể thao tác rất đơn giản, giống như jQuery vậy. Tuy nhiên, kết quả trả về của fetch là một Promise do đó các thao tác xử lý kết quả sẽ khác nhiều jQuery. *[Xem thêm nên học gì khi dùng jQuery.](https://topdev.vn/blog/hoc-gi-trong-jquery/)*

Lập trình với fetch rất đơn giản, thậm chí còn đơn giản hơn của với jQuery:

|  |  |
| --- | --- |
| 1  2  3  4  5 | fetch('https://jsonplaceholder.typicode.com/posts/1')  .then(response => response.json())  .then(console.log) |

**Cấu hình máy chủ hỗ trợ CORS**

Đây là phần phức tạp nhất, cũng là phần quan trọng nhất đối với CORS. Như đã nói ở trên, thực ra việc hỗ trợ CORS hay không phụ thuộc hoàn toàn vào máy chủ chứ không phải client.

Có hai loại truy vấn CORS: loại truy vấn “đơn giản” và “không đơn giản”.

Một truy vấn đơn giản hoàn toàn không cần đến CORS preflight. Một truy vấn sẽ được gọi là đơn giản nếu nó thoả mãn những điều kiện sau:

* Phương thức của truy vấn là một trong các loại GET, HEAD, POST.
* Giá trị của Content-Type phải là một trong số các loại application/x-www-form-urlencoded, multipart/form-data, text/plain.
* Không có event handler nào với event XMLHttpRequest.upload.
* Không sử dụng đối tượng ReadableStream trong truy vấn.
* Các HTTP header sau phải khớp:
  + Accept
  + Accept-Language
  + Content-Language
  + Last-Event-ID

Những truy vấn này được gọi là “đơn giản” bởi chúng có thể được coi là truy vấn thông thường từ trình duyệt mà không cần đến CORS, giống như submit một form HTML thông thường chẳng hạn.

Những truy vấn không phải “đơn giản” sẽ là truy vấn không đơn giản, và chúng cần CORS preflight. CORS preflight có nghĩa là trước khi truy vấn được gửi, nó cần phải gửi một truy vấn trước bằng phương thức OPTIONS. Mục đích của truy vấn “preflight” này là nhằm kiểm tra xem truy vấn thực sự có an toàn để gửi và nhận hay không.

Đối với truy vấn đơn giản

Một truy vấn CORS đơn giản như đã nói ở trên, có thể có gói tin HTTP dạng như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | GET /cors HTTP/1.1  Origin: https://api.topdevvn.com  Host: api.alice.com  Accept-Language: en-US  Connection: keep-alive  User-Agent: Mozilla/... |

Với các phương thức khác, gói tin HTTP cũng tương tự như vậy. Lưu ý rằng, một truy vấn CORS hợp lệ luôn luôn có Origin ở trong header. Giá trị của header này hoàn toàn được thiết lập tự động bởi trình duyệt, và không ai có thể thay đổi nó được. Giá trị của header này sẽ bao gồm scheme (http), domain (api.bob.com) và cổng (trong trường hợp dùng cổng mặc định thì không cần, ví dụ http dùng cổng 80). Giá trị của header chính là biểu thị nguồn gốc của truy vấn.

Một điểm lưu ý nữa là sự xuất hiện của header Origin không đồng nghĩa với việc truy vấn đó là cross origin. Dù tất cả các truy vấn cross origin đều có header này, nhưng một số truy vấn same origin cũng có header này. Điều đó phụ thuộc vào từng trình duyệt cụ thể.

Ví dụ, Firefox không có header này cho các truy vấn same origin nhưng Chrome và Safari vẫn thêm header nay khi truy vấn same origin nhưng sử dụng các phương thức POST, PUT hoặc DELETE. Đây là một diểm cần lưu ý với các [lập trình viên backend](https://topdev.vn/blog/roadmap-backend-developer/), vì nếu không để ý có thể không thêm origin của chính app trong danh sách các domain được chấp nhận, điều đó khiến cho chính truy vấn same origin lại gặp lỗi.

Dưới đây là response của máy chủ phản hồi cho một truy vấn CORS hợp lệ:

|  |  |
| --- | --- |
| 1  2  3  4  5 | Access-Control-Allow-Origin: https://api.topdevvn.com  Access-Control-Allow-Credentials: true  Access-Control-Expose-Headers: FooBar |

Tất cả các header liên quan đến CORS đều có phần đầu tiên là Acess-Control-. Ý nghĩa của từng header như sau:

* Access-Control-Allow-Origin (bắt buộc): đây là header phải có trong mọi response cho một truy vấn CORS hợp lệ. Nếu không có header này, truy vấn sẽ bị lỗi, giá trị của nó có thể là giá trị của header Origin được gửi lên hoặc \* biểu thị cho mọi origin.
* Access-Control-Allow-Credentials (tuỳ chọn): Mặc định, cookie sẽ không được sử dụng trong các truy vấn CORS. Header này sẽ biểu thị giá trị logic rằng có thể sử dụng cookie hay không. Giá trị duy nhất của header này là true. Nếu không muốn sử dụng cookie thì thông thường người ta sẽ bỏ header này trong response chứ không phải đặt giá trị nó là false. Lưu ý rằng, header này chỉ hoạt động nếu phía client cũng đặt giá trị withCredentials = true như đã nói ở phần trước.
* Access-Control-Expose-Headers (tuỳ chọn): Một đối tượng XMLHttpRequest có một phương thức getResponseHeader, phương thức này sẽ trả về giá trị của một header cụ thể trong response. Với các truy vấn CORS, phương thức này chỉ có thể truy cập được một số header đơn giản mà thôi. Nếu muốn phương thức này có thể truy cập nhiều header hơn, chúng ta cần đến giá trị của header này. Giá trị của header này là một danh sách các header có thể truy cập được, ngăn cách bằng dấu phẩy.

Đối với truy vấn cần preflight

Không phải truy vấn nào cũng là đơn giản do việc trao đổi dữ liệu giữa trình duyệt và máy chủ diễn ra rất đa dạng. Các phương thức PUT hay DELETE cũng thường xuyên được sử dụng. Ngoài ra kiểu dữ liệu [JSON](https://topdev.vn/blog/json-la-gi/) (Content-Type: application/json) cũng là lựa chọn của nhiều lập trình viên. Trong những trường hợp như vậy, trước khi truy vấn chính được thực hiện thì một truy vấn gọi là preflight sẽ được gửi đi trước.

Ở phía frontend, các truy vấn đơn giản hay phức tạp đều trông sẽ giống nhau. Truy vấn preflight hoàn toàn được thực hiện ngầm và trong suốt với người dùng. Truy vấn preflight sẽ được gửi đi trước nhằm xác định xem truy vấn thực sự có thể thực hiện được hay không.

Sau khi có được phản hồi tích cực, trình duyệt sẽ gửi truy vấn thực sự. Kết quả của truy vấn preflight có thể được cache nên nó không cần phải thực hiện cho mọi truy vấn.

**[Tuốt tuồn tuột về HTTP Polling và SSE (Server-sent event)](https://topdev.vn/blog/tuot-tuon-tuot-ve-http-polling-va-sse-server-sent-event/" \t "_blank)**

**[Cách làm HTTPS hoạt động trên local trong 5 phút](https://topdev.vn/blog/cach-lam-https-hoat-dong-tren-local-trong-5-phut/" \t "_blank)**

Dưới đây là một gói tin HTTP cho truy vấn preflight:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | OPTIONS /cors HTTP/1.1  Origin: https://api.topdevvn.com  Access-Control-Request-Method: PUT  Access-Control-Request-Headers: X-Custom-Header  Host: api.alice.com  Accept-Language: en-US  Connection: keep-alive  User-Agent: Mozilla/... |

Tương tự như truy vấn đơn giản, truy vấn này cũng tự động được thêm header Origin. Truy vấn preflight sẽ được thực hiện bằng phương thức OPTIONS với một số header đặc thù:

* Access-Control-Request-Method: Đây là phương thức HTTP dùng trong truy vấn thực sự. Giá trị của header luôn luôn phải có, ngay cả khi các phương thức đó cũng là phương thức của một truy vấn đơn giản.
* Access-Control-Request-Headers: Đây là danh sách (ngăn cách bằng dấu phẩy) các header được thêm vào truy vấn.

Truy vấn preflight là một cách để hỏi máy chủ rằng, liệu truy vấn thực sự có thể thực hiện được hay không. Mà máy chủ dựa vào hai header này để quyết định xem có chấp nhận truy vấn hay không. Nếu chấp nhận, máy chủ sẽ phản hồi như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | Access-Control-Allow-Origin: https://api.topdevvn.com  Access-Control-Allow-Methods: GET, POST, PUT  Access-Control-Allow-Headers: X-Custom-Header  Content-Type: text/html; charset=utf-8 |

Trong đó, response có thể có những header như sau:

* Access-Control-Allow-Origin (bắt buộc): Tương tự như trường hợp truy vấn CORS đơn giản.
* Access-Control-Allow-Methods (bắt buộc): Là một danh sách (ngăn cách bằng dấu phẩy) các phương thức HTTP được chấp nhận. Dù truy vấn preflight có hỏi về một phương thức cụ thể của truy vấn tiếp theo, giá trị của header này trong responses có thể bao gồm tất cả các phương thức được chấp nhận.
* Access-Control-Allow-Headers (bắt buộc nếu truy vấn có header Access-Control-Request-Headers): Là danh sách các header (ngăn cách bằng dấu phẩy) được hỗ trợ. Tương tự như header trước, giá trị của header này cũng có thể bao gồm tất cả các header được chấp nhận.
* Access-Control-Allow-Credentials (tuỳ chọn): Tương tự như trường hợp truy vấn CORS đơn giản.
* Access-Control-Max-Age (tuỳ chọn): Truy vấn preflight không nhất thiết phải được thực hiện cho mọi truy vấn, mà kết quả của nó có thể cache được. Giá trị của header này chính là số giây mà giá trị của truy vấn preflight có thể được cache.

Một khi truy vấn preflight có được phản hồi và được chấp nhận, trình duyệt sẽ thực hiện truy vấn thực sự. Truy vấn lúc này tương tự như truy vấn CORS đơn giản và quá trình xử lý cũng như phản hồi hoàn toàn tương tự như vậy.

Nếu muốn từ chối truy vấn CORS, máy chủ có thể phần hồi một gói tin HTTP bình thường (mã 200) nhưng không có chứa HTTP header nào liên quan đến CORS. Trong trường hợp truy vấn preflight nhận được phản hồi như vậy, trình duyệt sẽ hiểu là truy vấn không được chấp nhận và nó sẽ không gửi thêm truy vấn nào nữa.

Về phía client, nếu trong trường hợp không thực hiện được truy vấn, event onerror sẽ được gọi. Tuy nhiên, như đã nó ở trên, trình duyệt cũng không thể truy cập được nhiều thông tin về lỗi đó, chỉ đơn giản là biết có lỗi mà thôi.

**Hỗ trợ CORS của các framework**

**Laravel CORS**

Khi chúng ta code vài ứng dụng dưới local mà có connect tới Laravel backed, thì bạn sẽ nhận cái thông báo error CORS ngay. Vì vậy cần tạo một [middleware](https://topdev.vn/blog/middleware-la-gi/) sau:

|  |  |
| --- | --- |
| 1  2  3 | $ php artisan make:middleware Cors |

Sau đó update header trong **app/Http/Middleware/Cors.php**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <?php  namespace App\Http\Middleware;  use Closure;  class Cors  {    public function handle($request, Closure $next)    {      return $next($request)        ->header(‘Access-Control-Allow-Origin’, ‘\*’)        ->header(‘Access-Control-Allow-Methods’, ‘GET, POST, PUT, DELETE, OPTIONS’)        ->header(‘Access-Control-Allow-Headers’, ‘X-Requested-With, Content-Type, X-Token-Auth, Authorization’);    }  } |

Sau đó, đăng ký middleware trong **app/Http/kernel.php**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | protected $routeMiddleware = [    ‘auth’ => \App\Http\Middleware\Authenticate::class,    ‘auth.basic’ => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,    ‘guest’ => \App\Http\Middleware\RedirectIfAuthenticated::class,    ‘cors’ => \App\Http\Middleware\Cors::class, // <-- thêm vào hàng này  ]; |

Nếu bạn muốn bật CORS bất kỳ route nào, chỉ cần add middleware này trong route registration. Ngoài ra bạn có thể dùng packgage ngoài tại [barryvdh/](https://github.com/barryvdh/laravel-cors" \t "_self)**[laravel-cors](https://github.com/barryvdh/laravel-cors" \t "_self)**.

**CORS Nodejs**

Chúng ta sử dụng dòng code dưới đây để set một header trên response của bạn để bật CORS:

|  |  |
| --- | --- |
| 1  2  3 | res.header("Access-Control-Allow-Origin", "\*"); |

Bật CORS cho toàn bộ resource trên server

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | app.use(function(req, res, next) {    res.header("Access-Control-Allow-Origin", "\*");    res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");    next();  }); |

Chỉ định file nào đó

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | app.get('/file', function(req, res){    var file = \_\_dirname + '/file.zip';    res.download(file); // Set disposition and send it.  }); |

Ví dụ đầy đủ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | var express = require('express');  var app = express();  app.use(function(req, res, next) {    res.header("Access-Control-Allow-Origin", "\*");    res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");    next();  });  app.get('/', function (req, res) {    var data = {      "SmartPhone": [        "iPhone",        "Samsung"    ]    };    res.json(data);  });  app.get('/file', function(req, res){    var file = \_\_dirname + '/file.zip';    res.download(file); // Set disposition and send it.  }); |

**Ruby on Rails**

Ruby on Rails cho phép chúng ta thiết lập cũng như thay đổi các header trong response khá dễ dàng, do đó, muốn chấp nhận truy vấn CORS, chúng ta có thể đơn giản là làm như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | skip\_before\_filter :verify\_authenticity\_token  before\_action :cors\_preflight\_check  after\_action :cors\_set\_access\_control\_headers    def cors\_set\_access\_control\_headers    headers["Access-Control-Allow-Origin"] = "\*"    headers["Access-Control-Allow-Methods"] = "GET, POST"    headers["Access-Control-Max-Age"] = "1728000"  end    def cors\_preflight\_check    headers["Access-Control-Allow-Origin"] = "\*"    headers["Access-Control-Allow-Methods"] = "GET, POST"    headers["Access-Control-Allow-Headers"] =      "X-Requested-With, X-Prototype-Version"    headers["Access-Control-Max-Age"] = "1728000"  end |

**Django**

Với Django chúng ta phải sử dụng thêm một package, đó là [Django CORS headers](https://github.com/ottoyiu/django-cors-headers). Package này sẽ giúp chúng ta thiết lập các header cần thiết cho một truy vấn CORS, đồng thời cho chúng ta khả năng cấu hình URL nào cho phép CORS, URL nào thì không.

Với package này, chúng ta có thể cấu hình sao cho chỉ có API mới hỗ trợ CORS như sau:

|  |  |
| --- | --- |
| 1  2  3  4 | CORS\_ORIGIN\_ALLOW\_ALL = True  CORS\_URLS\_REGEX = r'^/api/v1/.\*$' |

Ngoài ra còn rất nhiều cấu hình khác nữa, cho phép chúng ta chỉ chấp nhận truy vấn CORS từ một vài origin nhất định chẳng hạn (CORS\_ORIGIN\_REGEX\_WHITELIST). Nội dung chi tiết xin mới các bạn xem cụ thể tại README của package đó.

**Flask**

Tương tự như Django, với Flask, chúng ta cũng phải sử dụng thêm một package, đó là [Flask-CORS](https://flask-cors.readthedocs.io/en/latest/) mới có thể chấp nhận các truy vấn CORS được.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | from flask import Flask  from flask.ext.cors import CORS, cross\_origin      app = Flask(\_\_name\_\_)  app.config['SECRET\_KEY'] = 'The quick brown fox jumps over the lazy dog'  app.config['CORS\_HEADERS'] = 'Content-Type'    cors = CORS(app, resources={r'/foo/\*': {'origins': '\*'}})      @app.route('/foo')  @cross\_origin()  def foo():  return 'Hello, world! CORS works'      if \_\_name\_\_ == '\_\_main\_\_':  app.run() |