الجمهورية العربية السورية
جامعة تشرين
كلية الهندسة الميكانيكية والكهربائية
قسم هندسة الاتصالات والالكترونيات
برمجة شبكات

إنشاء خادم HTTP بسيط وعميل فيpython

بروتوكول HTTP (Hypertext Transfer Protocol)

راما خالد حداد 2059

حلا محمود علي 2273

ملخص:

يعمل HTTP كبروتوكول استجابة للطلب في نموذج الخادم والعميل. قد يكون مستعرض الويب، على سبيل المثال، هو العميل، في حين أن عملية، تسمى خادم الويب، تعمل على جهاز كمبيوتر يستضيف موقع ويب واحدًا أو أكثر قد تكون هي الخادم. يرسل العميل رسالة طلب HTTP إلى الخادم. يقوم الخادم، الذي يوفر موارد مثل ملفات HTMLوالمحتويات الأخرى أو يؤدي وظائف أخرى نيابة عن العميل، بإرجاع رسالة استجابة إلى العميل. تحتوي الاستجابة على معلومات حالة الإكمال حول الطلب وقد تحتوي أيضًا على المحتوى المطلوب في نص رسالتها.

متصفح الويب هو مثال على وكيل المستخدم .(UA) تشمل الأنواع الأخرى من وكلاء المستخدم برنامج الفهرسة الذي يستخدمه موفرو البحث (برامج زحف الويب) والمتصفحات الصوتية وتطبيقات الأجهزة المحمولة والبرامج الأخرى التي تصل إلى محتوى الويب أو تستهلكه أو تعرضه.

تم تصميم HTTP للسماح لعناصر الشبكة الوسيطة بتحسين أو تمكين الاتصالات بين العملاء والخوادم. غالبًا ما تستفيد مواقع الويب عالية الحركة من خوادم ذاكرة التخزين المؤقت على الويب التي تقدم المحتوى نيابة عن الخوادم الأولية لتحسين وقت الاستجابة. تقوم متصفحات الويب بالتخزين المؤقت لموارد الويب التي تم الوصول إليها مسبقًا وإعادة استخدامها، كلما أمكن ذلك، لتقليل حركة مرور الشبكة. يمكن لخوادم بروكسي HTTP على حدود الشبكة الخاصة تسهيل الاتصال للعملاء بدون عنوان قابل للتوجيه عالميًا، عن طريق ترحيل الرسائل مع خوادم خارجية.

كلمات مفتاحية: HTTP

Abstract: HTTP acts as a request response protocol in the client-server model. server server server server email server. The client sends an HTTP request message to the server. The server provides resources such as HTML files and other content or performs other functions on behalf of the client, returning a receiving message to the client. It contains information about the completion status, and may also contain the requested content in the body of its message.

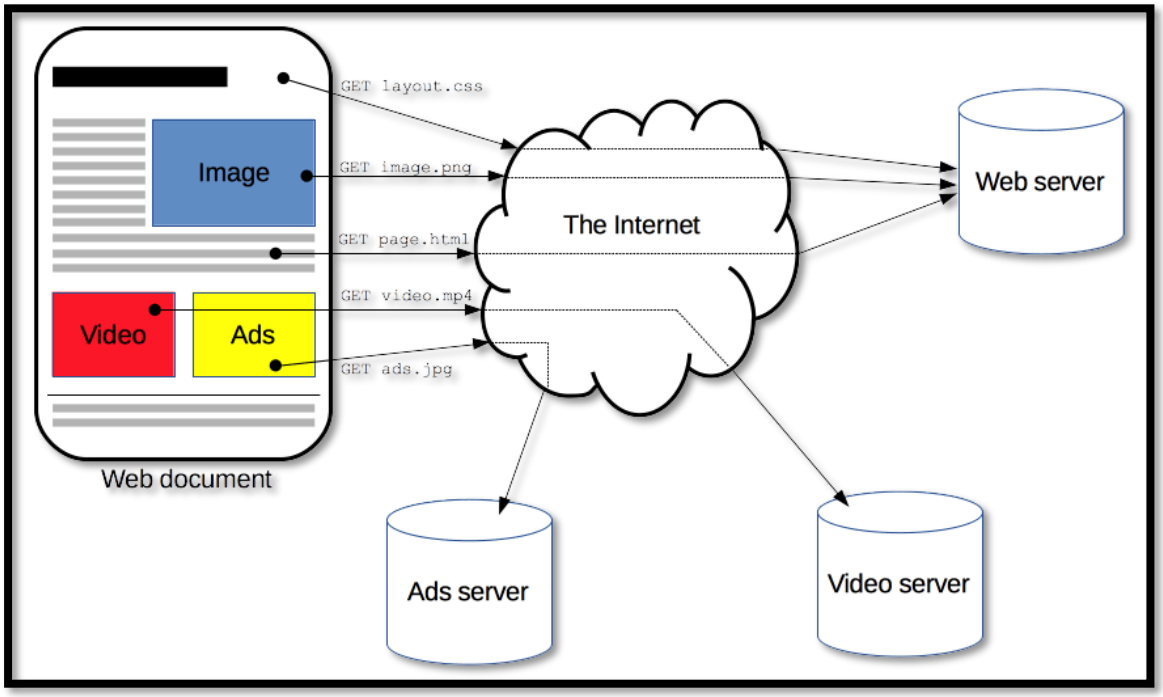
A web browser is an example of a user agent (UA). Other factors include proxies for the indexing software used by search providers (web crawlers) and browsers, applications, software, and other applications that access, consume, or display web content.

HTTP Intermediate network elements are designed to improve or improve communications between clients and servers. Take advantage of the email office cache. You are restocking email. It can be for proxy servers, HTTP on private network borders, within Saudi Arabia, and mobile.

Keywords: HTTP

مقدمة

HTTP هو بروتوكول لجلب الموارد مثل مستندات HTML. إنه أساس أي تبادل بيانات على الويب وهو بروتوكول خادم عميل، مما يعني أن الطلبات تبدأ من قبل المستلم، وعادة ما يكون مستعرض الويب. يتم إعادة بناء المستند الكامل من المستندات الفرعية المختلفة التي تم جلبها، على سبيل المثال، النص ووصف التخطيط والصور ومقاطع الفيديو والنصوص والمزيد. كما يوضح في الشكل (1).



الشكل (1)

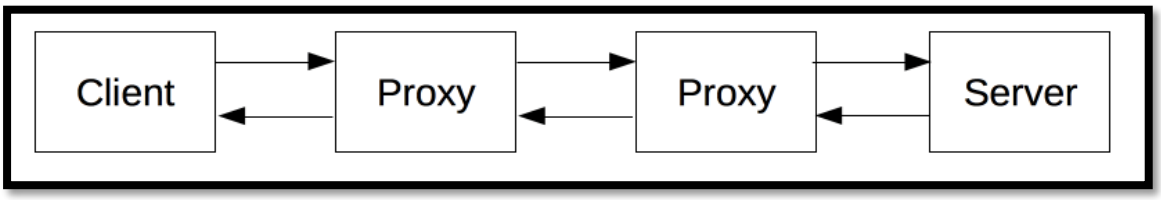
يتواصل العملاء والخوادم من خلال تبادل الرسائل الفردية (بدلاً من تدفق البيانات). الرسائل المرسلة من قبل العميل، وعادة ما تكون مستعرض ويب، تسمى الطلبات والرسائل التي يرسلها الخادم كإجابة تسمى الردود.

تم تصميم HTTP في أوائل التسعينيات، وهو بروتوكول قابل للتوسيع تطور بمرور الوقت. إنه بروتوكول طبقة تطبيق يتم إرساله عبر TCP، أو عبر اتصال TCP مشفر بواسطة TLS، على الرغم من إمكانية استخدام أي بروتوكول نقل موثوق به نظريًا. نظرًا لقابليته للتوسعة، فإنه لا يستخدم فقط لجلب مستندات النص التشعبي، ولكن أيضًا الصور ومقاطع الفيديو أو لنشر المحتوى على الخوادم، كما هو الحال مع نتائج نماذج HTML. يمكن أيضًا استخدام HTTP لجلب أجزاء من المستندات لتحديث صفحات الويب عند الطلب.

1. مكونات الأنظمة المستندة إلى HTTP:

HTTPهو بروتوكول خادم عميل: يتم إرسال الطلبات بواسطة كيان واحد ، وكيل المستخدم (أو وكيل نيابة عنه). في معظم الأحيان يكون وكيل المستخدم مستعرض ويب، ولكن يمكن أن يكون أي شيء، على سبيل المثال، روبوت يزحف إلى الويب لملء فهرس محرك البحث والحفاظ عليه.

يتم إرسال كل طلب فردي إلى الخادم الذي يتعامل معه ويقدم إجابة تسمى الاستجابة. يوجد بين العميل والخادم العديد من الكيانات، والتي تسمى مجتمعة البروكسيات، والتي تؤدي عمليات مختلفة وتعمل كبوابات أو مخابئ، على سبيل المثال ونوضح ذلك بالشكل (2).



الشكل (2).

في الواقع، يوجد عدد أكبر من أجهزة الكمبيوتر بين المتصفح والخادم الذي يتعامل مع الطلب: هناك أجهزة توجيه ومودم وغير ذلك. بفضل التصميم متعدد الطبقات للويب، يتم إخفاؤها في طبقات الشبكة والنقل. HTTP في القمة ، في طبقة التطبيق. على الرغم من أهميتها لتشخيص مشاكل الشبكة، إلا أن الطبقات الأساسية لا علاقة لها في الغالب بوصف HTTP.

1.1 . العميل: وكيل المستخدم:

وكيل المستخدم هو أي أداة تعمل نيابة عن المستخدم. يتم تنفيذ هذا الدور بشكل أساسي بواسطة مستعرض الويب، ولكن يمكن أيضًا تنفيذه بواسطة البرامج المستخدمة من قبل المهندسين ومطوري الويب لتصحيح أخطاء تطبيقاتهم.

المستعرض هو دائمًا الكيان الذي يبدأ الطلب. إنه ليس الخادم أبدًا (على الرغم من إضافة بعض الآليات على مر السنين لمحاكاة الرسائل التي يبدأها الخادم).

لعرض صفحة ويب، يرسل المستعرض طلبًا أصليًا لجلب مستند HTML الذي يمثل الصفحة. يقوم بعد ذلك بتحليل هذا الملف، وتقديم طلبات إضافية تتوافق مع نصوص التنفيذ، ومعلومات التخطيط (CSS) لعرضها، والموارد الفرعية

الموجودة في الصفحة (عادةً الصور ومقاطع الفيديو). يقوم مستعرض الويب بعد ذلك بدمج هذه الموارد لتقديم المستند الكامل، صفحة الويب. يمكن أن تجلب البرامج النصية التي ينفذها المتصفح مزيدًا من الموارد في مراحل لاحقة ويقوم المتصفح بتحديث صفحة الويب وفقًا لذلك.

صفحة الويب هي مستند نص تشعبي. هذا يعني أن بعض أجزاء المحتوى المعروض عبارة عن روابط يمكن تنشيطها (عادةً بنقرة على الماوس) لجلب صفحة ويب جديدة، مما يسمح للمستخدم بتوجيه وكيل المستخدم الخاص به والتنقل عبر الويب. يترجم المتصفح هذه التوجيهات إلى طلبات HTTP، ويفسر كذلك استجابات HTTP لتقديم استجابة واضحة للمستخدم.

2.1. خادم الويب:

على الجانب الآخر من قناة الاتصال يوجد الخادم، الذي يخدم المستند حسب طلب العميل. يظهر الخادم كآلة واحدة فقط تقريبًا؛ ولكنها قد تكون في الواقع مجموعة من الخوادم التي تشترك في الحمل (موازنة التحميل)، أو قطعة معقدة من البرامج تستجوب أجهزة الكمبيوتر الأخرى (مثل ذاكرة التخزين المؤقت، أو خادم قاعدة البيانات، أو خوادم التجارة الإلكترونية)، مما يؤدي إلى إنشاء المستند كليًا أو جزئيًا عند الطلب.

لا يكون الخادم بالضرورة جهازًا واحدًا، ولكن يمكن استضافة العديد من مثيلات برنامج الخادم على نفس الجهاز. باستخدام HTTP / 1.1 ورأس المضيف، يمكنهم أيضًا مشاركة عنوان IP نفسه.

3.1. الوكلاء:

بين متصفح الويب والخادم، تقوم العديد من أجهزة الكمبيوتر والآلات بترحيل رسائل HTTP .نظرًا للهيكل متعدد الطبقات لمكدس الويب، يعمل معظمها على مستوى النقل أو الشبكة أو المستوى المادي، وتصبح شفافة في طبقة HTTPويحتمل أن يكون لها تأثير كبير على الأداء. تسمى تلك التي تعمل في طبقات التطبيق عمومًا الوكلاء.

يمكن أن تكون هذه شفافة، حيث يتم إعادة توجيه الطلبات التي يتلقونها دون تغييرها بأي شكل من الأشكال، أو غير شفافة، وفي هذه الحالة سوف يقومون بتغيير الطلب بطريقة ما قبل تمريره إلى الخادم. قد تؤدي الوكلاء وظائف عديدة:

التخزين المؤقت (يمكن أن تكون ذاكرة التخزين المؤقت عامة أو خاصة، مثل ذاكرة التخزين المؤقت للمتصفح)

التصفية (مثل فحص مكافحة الفيروسات أو المراقبة الأبوية)

موازنة الحمل (للسماح لخوادم متعددة بخدمة طلبات مختلفة)

المصادقة (للتحكم في الوصول إلى الموارد المختلفة)

التسجيل (السماح بتخزين المعلومات التاريخية)

2. الجوانب الأساسية لـ HTTP:

1.2. HTTP بسيط:

تم تصميم HTTP بشكل عام ليكون بسيطًا وقابل للقراءة البشرية، حتى مع التعقيد الإضافي المقدم في HTTP / 2 عن طريق تغليف رسائل HTTP في إطارات. يمكن قراءة رسائل HTTP وفهمها من قبل البشر، مما يوفر اختبارًا أسهل للمطورين، ويقلل من التعقيد للقادمين الجدد.

2.2.HTTP قابل للتوسيع:

المقدمة في HTTP / 1.0 ، تجعل رؤوس HTTP هذا البروتوكول سهل التوسيع والتجربة. يمكن أيضًا تقديم وظائف جديدة من خلال اتفاقية بسيطة بين العميل والخادم حول دلالات الرأس الجديد.

3.2.HTTP عديم الحالة ، ولكن ليس بدون جلسات:

HTTP عديم الحالة: لا يوجد ارتباط بين طلبين يتم تنفيذهما على التوالي على نفس الاتصال. هذا على الفور لديه احتمال أن يكون مشكلة للمستخدمين الذين يحاولون التفاعل مع صفحات معينة بشكل متماسك، على سبيل المثال، باستخدام سلال التسوق للتجارة الإلكترونية. ولكن في حين أن جوهر HTTP نفسه عديم الحالة، فإن ملفات تعريف

ارتباط HTTP تسمح باستخدام الجلسات ذات الحالة. باستخدام قابلية التوسعة، تتم إضافة ملفات تعريف ارتباط HTTP إلى سير العمل، مما يسمح بإنشاء الجلسة على كل طلب HTTP لمشاركة نفس السياق، أو نفس الحالة.

HTTP.4.2 والاتصالات:

يتم التحكم في الاتصال في طبقة النقل، وبالتالي فهو في الأساس خارج نطاق HTTP. لا يتطلب HTTP أن يكون بروتوكول النقل الأساسي قائماً على الاتصال؛ يتطلب فقط أن تكون موثوقة، أو لا تفقد الرسائل (على الأقل، تقديم خطأ في مثل هذه الحالات). من بين بروتوكولي النقل الأكثر شيوعًا على الإنترنت، يعتبر TCP موثوقًا و UDP ليس كذلك. لذلك يعتمد HTTP على معيار TCP ، الذي يعتمد على التوصيل.

قبل أن يتمكن العميل والخادم من تبادل زوج طلب / استجابةHTTP ، يجب عليهم إنشاء اتصالTCP ، وهي عملية تتطلب عدة رحلات ذهابًا وإيابًا. السلوك الافتراضي لـ HTTP / 1.0 هو فتح اتصال TCP منفصل لكل زوج من طلبات / استجابة .HTTP. هذا أقل كفاءة من مشاركة اتصال TCP واحد عندما يتم إرسال طلبات متعددة في تتابع قريب.

من أجل التخفيف من هذا الخلل، قدم HTTP / 1.1 خطوط الأنابيب (التي ثبت صعوبة تنفيذها) والتوصيلات المستمرة: يمكن التحكم في اتصال TCP الأساسي جزئيًا باستخدام رأس الاتصال. ذهب HTTP / 2 خطوة إلى الأمام من خلال تعدد إرسال الرسائل عبر اتصال واحد، مما يساعد في الحفاظ على الاتصال دافئًا وأكثر كفاءة.

5.2. ما يمكن التحكم فيه عن طريق HTTP

هذه الطبيعة القابلة للتوسيع لـHTTP ، بمرور الوقت، سمحت بمزيد من التحكم والوظائف على الويب. تم التعامل مع أساليب التخزين المؤقت والمصادقة في وقت مبكر في محفوظات .HTTP. على النقيض من ذلك، تمت إضافة القدرة على تخفيف قيود الأصل فقط في 2010.

فيما يلي قائمة بالميزات الشائعة التي يمكن التحكم فيها باستخدام:HTTP

يمكن التحكم في كيفية تخزين المستندات مؤقتًا بواسطة .HTTP يمكن للخادم إرشاد الوكلاء والعملاء حول ما سيتم تخزينه مؤقتًا ومدة ذلك. يمكن للعميل توجيه وكلاء ذاكرة التخزين المؤقت الوسيطة لتجاهل المستند المخزن.

تخفيف قيود الأصل لمنع التطفل وانتهاكات الخصوصية الأخرى، تفرض مستعرضات الويب فصلًا صارمًا بين مواقع الويب. يمكن فقط للصفحات من نفس الأصل الوصول إلى كافة المعلومات الخاصة بصفحة الويب. على الرغم من أن مثل هذا القيد يمثل عبئًا على الخادم، إلا أن رؤوس HTTP يمكن أن تخفف من هذا الفصل الصارم على جانب الخادم، مما يسمح للمستند بأن يصبح خليطًا من المعلومات التي يتم الحصول عليها من مجالات مختلفة؛ قد تكون هناك أسباب أمنية للقيام بذلك.

المصادقة قد تكون بعض الصفحات محمية بحيث يمكن لمستخدمين معينين فقط الوصول إليها. قد يتم توفير المصادقة الأساسية عن طريقHTTP ، إما باستخدام مصادقة WWW ورؤوس مماثلة، أو عن طريق تعيين جلسة محددة باستخدام ملفات تعريف ارتباط.HTTP

غالبًا ما توجد الخوادم الوكيلة والنفق أو العملاء على شبكات الإنترانت وتخفي عنوان IP الحقيقي الخاص بهم من أجهزة الكمبيوتر الأخرى. تنتقل طلبات HTTP بعد ذلك عبر الوكلاء لعبور حاجز الشبكة هذا. ليست كل الوكلاء عبارة عن وكلاء .HTTP بروتوكولSOCKS ، على سبيل المثال، يعمل بمستوى أدنى. يمكن معالجة البروتوكولات الأخرى، مثل بروتوكول نقل الملفات، بواسطة هذه البروتوكولات.

الجلسات يتيح لك استخدام ملفات تعريف الارتباط HTTP ربط الطلبات بحالة الخادم. يؤدي هذا إلى إنشاء جلسات، على الرغم من أن HTTP الأساسي هو بروتوكول بدون حالة. هذا مفيد ليس فقط لسلال التسوق في التجارة الإلكترونية، ولكن أيضًا لأي موقع يسمح للمستخدم بتكوين المخرجات.

6.2. تدفق HTTP

عندما يريد العميل الاتصال بخادم، إما الخادم النهائي أو الوكيل الوسيط، فإنه يقوم بالخطوات التالية:

- فتح اتصال :TCP يُستخدم اتصال TCP لإرسال طلب أو عدة طلبات والحصول على إجابة. يجوز للعميل فتح اتصال جديد أو إعادة استخدام اتصال موجود أو فتح عدة اتصالات TCP للخوادم.
- إرسال رسالة HTTP: رسائل HTTP (قبل HTTP / 2) قابلة للقراءة من قبل الإنسان. باستخدام HTTP 2 /، يتم تغليف هذه الرسائل البسيطة في إطارات، مما يجعل من المستحيل قراءتها مباشرةً، ولكن يظل المبدأ كما هو.
- قراءة الرد الذي أرسله الخادم
- أغلاق الاتصال.

إذا تم تنشيط تدفق HTTP، فيمكن إرسال العديد من الطلبات دون انتظار الرد الأول ليتم استلامه بالكامل. ثبت أن تنفيذ خطوط أنابيب HTTP صعب التنفيذ في الشبكات الحالية، حيث تتعايش الأجزاء القديمة من البرامج مع الإصدارات الحديثة. تم استبدال خطوط أنابيب HTTP في HTTP / 2 بطلبات تعدد إرسال أكثر قوة داخل إطار.

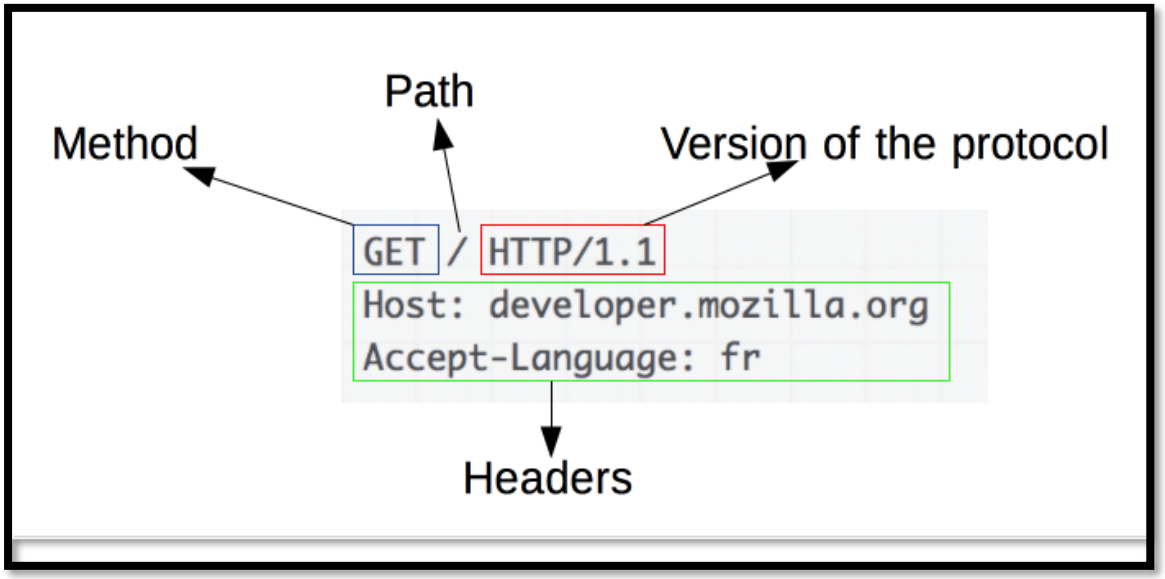
7.2. رسائل HTTP

رسائلHTTP ، على النحو المحدد في HTTP / 1.1 والإصدارات الأقدم، يمكن قراءتها من قبل الإنسان. في HTTP / 2، يتم تضمين هذه الرسائل في هيكل ثنائي، إطار، مما يسمح بالتحسينات مثل ضغط الرؤوس وتعدد الإرسال. حتى إذا تم إرسال جزء فقط من رسالة HTTP الأصلية في هذا الإصدار منHTTP ، فإن دلالات كل رسالة لا تتغير ويعيد العميل (افتراضيًا) تكوين طلب HTTP / 1.1 الأصلي. لذلك من المفيد فهم رسائل HTTP 2 /بتنسيق.HTTP / 1.1

هناك نوعان من رسائل HTTP والطلبات والاستجابات، ولكل منهما تنسيقه الخاص.

الطلبات

مثال لطلب HTTP:



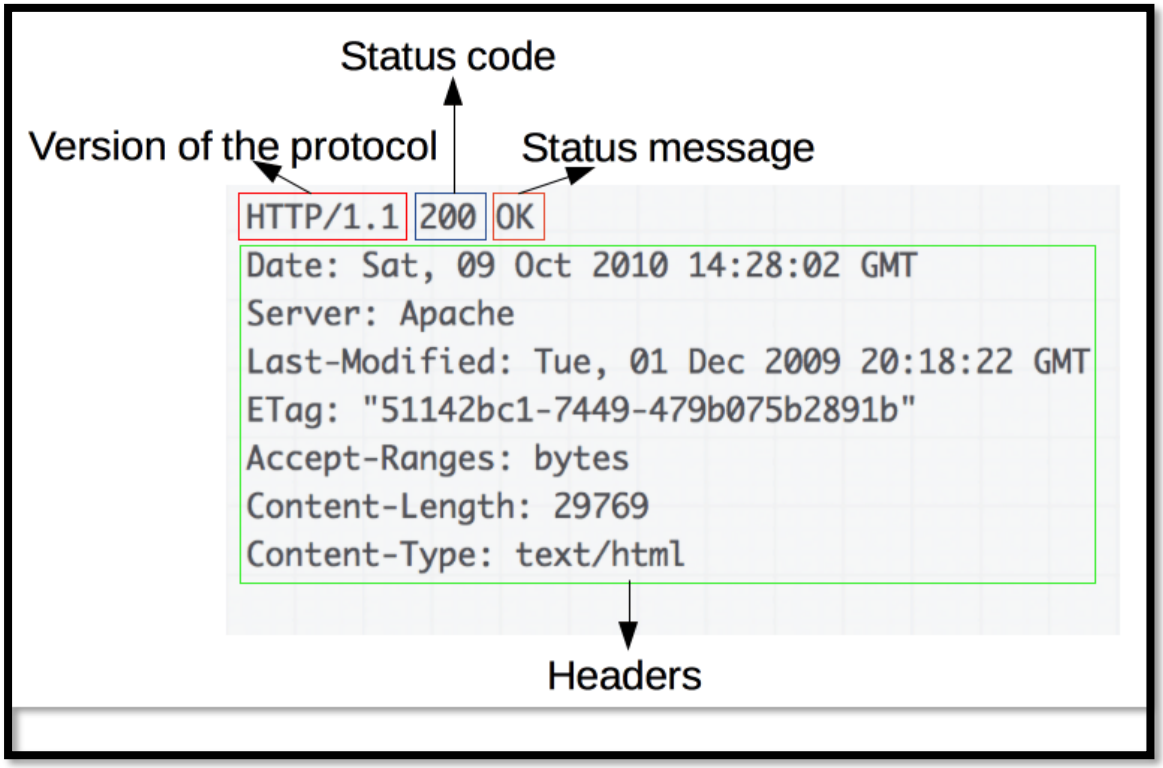
الشكل (3).

تتكون الطلبات من العناصر التالية

طريقةHTTP ، عادةً ما تكون فعلًا مثل GET أو POST أو اسمًا مثل OPTIONS أو HEAD والتي تحدد العملية التي يريد العميل تنفيذها. عادةً ما يريد العميل جلب مورد باستخدام GET أو نشر قيمة نموذج HTML باستخدامPOST على الرغم من أنه قد تكون هناك حاجة لمزيد من العمليات في حالات أخرى.

مسار المورد المراد جلبه؛ عنوان URL للمورد تم تجريده من العناصر الواضحة من السياق، على سبيل المثال بدون البروتوكول (http: //) أو المجال) هنا، developer.mozilla.orgأو منفذ) TCP هنا، 80.(إصدار بروتوكول HTTP.

رد على سبيل المثال:



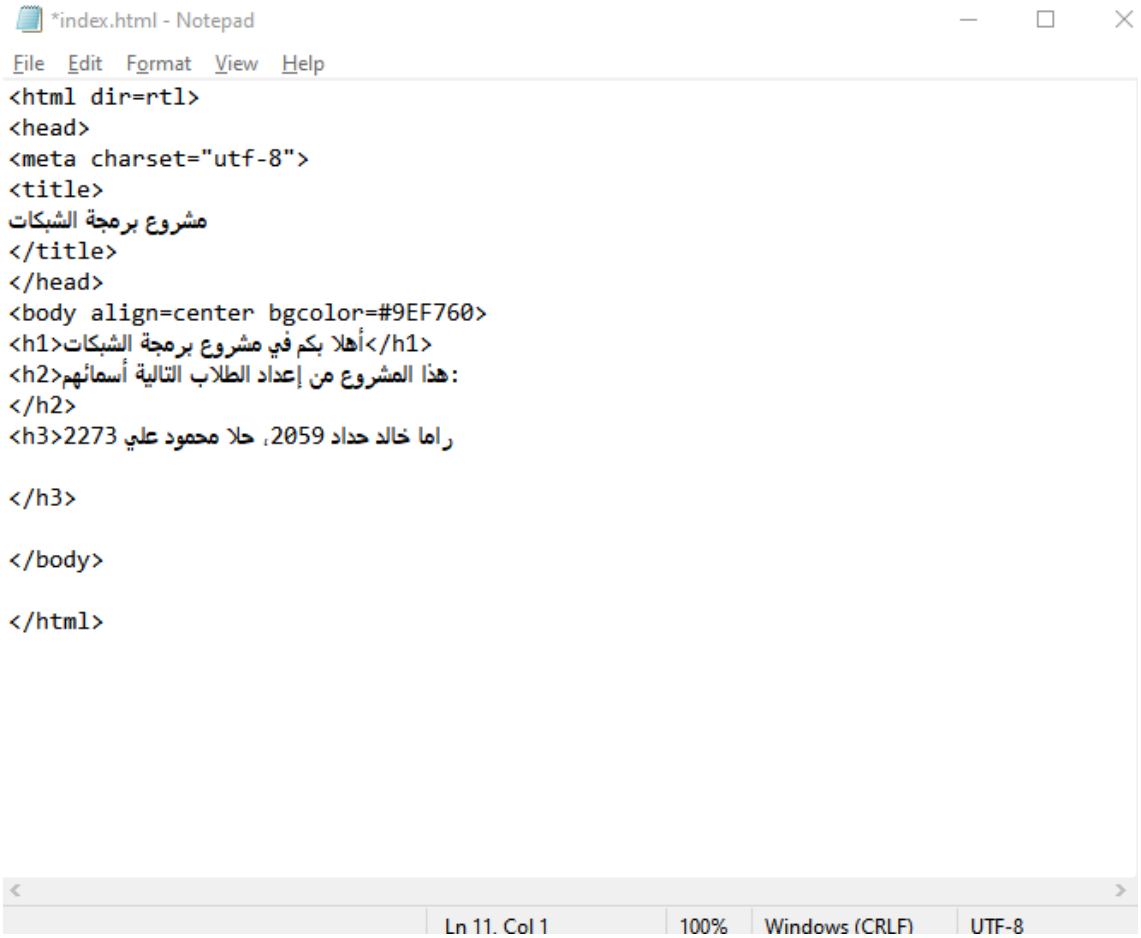
الشكل (4).

تتكون الردود من العناصر التالية:

- إصدار بروتوكول HTTP الذي يتبعونه.
- رمز حالة، يوضح ما إذا كان الطلب ناجحًا أم لا، وسبب ذلك.
- رسالة حالة، وصف قصير غير موثوق لرمز الحالة.
- رؤوسHTTP ، مثل تلك الخاصة بالطلبات.
- اختياريًا، جسم يحتوي على المورد الذي تم جلبه.

1.4. سيناريو المحاكاة:

نقوم بإنشاء مجلد خاص بالمشروع في القرص c ونسميه http مثلاً وبعدها نقوم بإنشاء ملف html باستخدام المفكرة وتغيير صيغة الملف من txt إلى html وكتابة كود الـ html ضمن الملف ويجب أن يكون اسم الملف index.html ليفهم السيرفر أن هذا الملف هو الصفحة الرئيسية وبالتالي عند الاتصال بالسيرفر سيعيد لنا هذه الصفحة ليتم عرضها على متصفح الويب.



نقوم بإنشاء سيرفر http ضمن نفس المجلد وذلك بالانتقال إلى المجلد عن طريق موجه الاوامر وكتابة التعليمة التالية:

python –m http.server

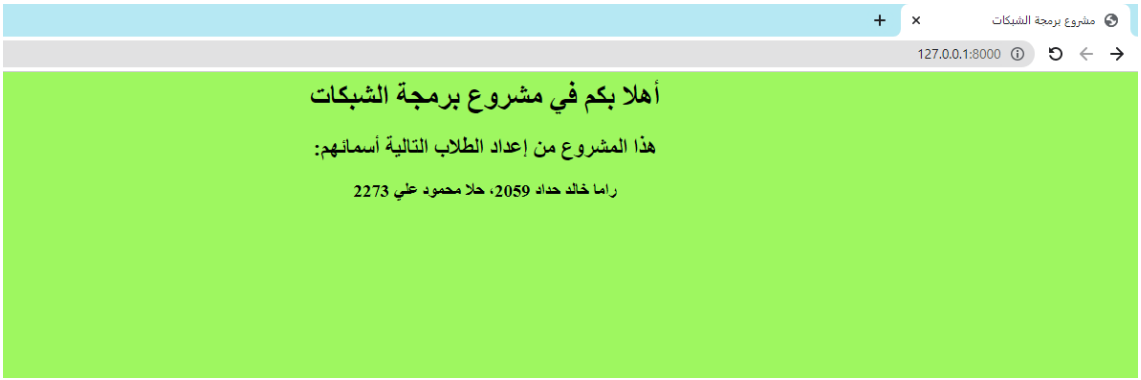
كما هو موضح في الشكل:

```
c:\http>python -m http.server
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
```

الشكل (5).

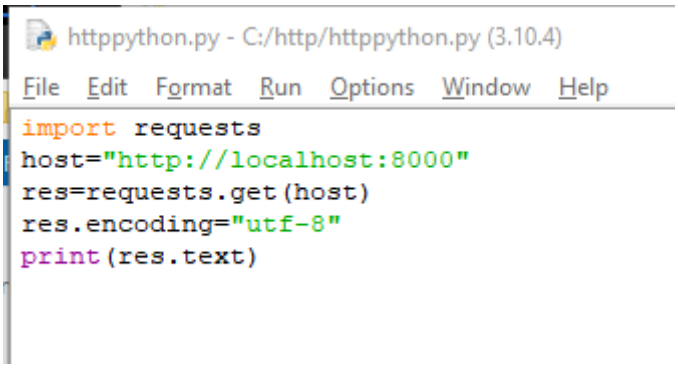
فيتم إنشاء http Server يعمل على ip الجهاز ويتنصت على الـ port 8000 في هذه الحالة يكون الـ ip هو localhost أو 127.0.0.1

الآن أصبح لدينا سيرفر http بسيط يعمل على البورت 8000 وبالتالي نقوم بالاتصال به عن طريق الزبون وهو متصفح chrome مثلاً



الشكل (6).

الآن سنقوم بالاتصال عبر البايثون باستخدام مكتبة الـ requests من خلال الميثود get كما هو موضح بالكود:



الشكل (7).

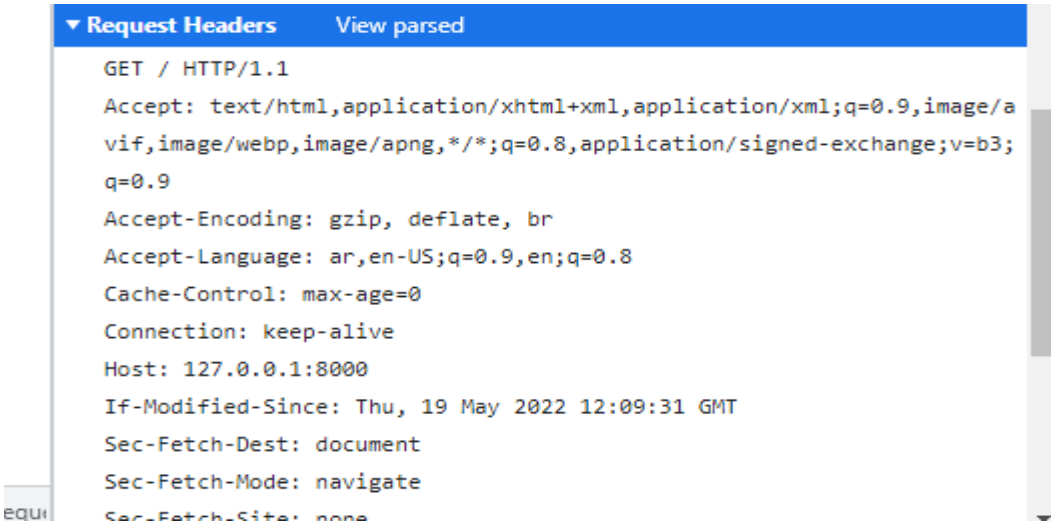
2.4. النتائج والمناقشة:

يظهر على موجه الأوامر التوابع المستخدمة وعدد مرات استخدامها ونلاحظ استخدام التابع GET في كل مرة أقوم فيها بالاتصال بالسيرفر إن كان عبر المتصفح أو عبر المكتبة requests :

```
c:\http>python -m http.server
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
::ffff:127.0.0.1 - - [19/May/2022 15:28:24] "GET / HTTP/1.1" 200 -
::ffff:127.0.0.1 - - [19/May/2022 15:28:24] code 404, message File not found
::ffff:127.0.0.1 - - [19/May/2022 15:28:24] "GET /favicon.ico HTTP/1.1" 404 -
::ffff:127.0.0.1 - - [19/May/2022 16:00:22] "GET / HTTP/1.1" 304 -
::ffff:127.0.0.1 - - [19/May/2022 16:00:23] "GET / HTTP/1.1" 304 -
::1 - - [19/May/2022 16:01:21] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:01:30] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:01:38] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:02:02] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:02:20] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:02:32] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:03:45] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:03:55] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:04:07] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:05:53] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:06:15] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:06:27] "GET / HTTP/1.1" 200 -
::1 - - [19/May/2022 16:07:24] "GET / HTTP/1.1" 200 -
```

الشكل (8).

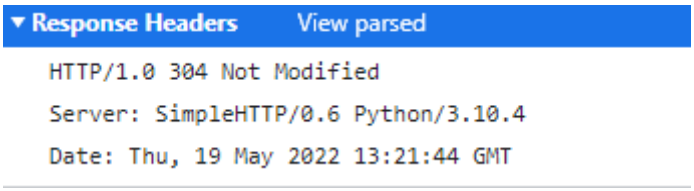
نعرض شكل الطلب من خلال المتصفح كما هو موضح بالشكل:



الشكل (9).

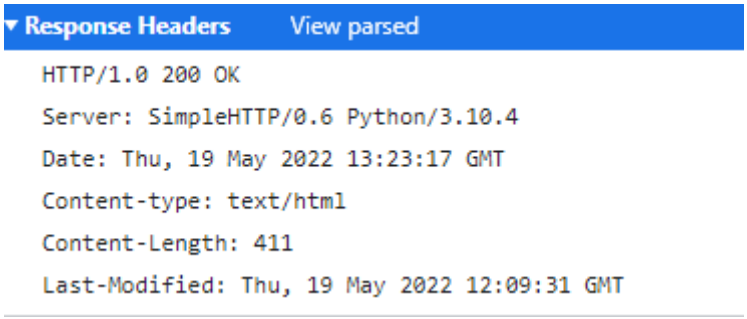
حيث التابع المستخدم هو GET والبروتوكول HTTP وإصداره 1.1

يكون شكل الاستجابة:



الشكل (10).

الكود الخاص بالاستجابة هو 304 أي قام بإعادة تحويل الطلب إلى العنوان localhost:8000 ولو قمت بطلب هذا العنوان من المتصفح سيكون الكود هو 200 أي تمت الاستجابة بشكل صحيح:



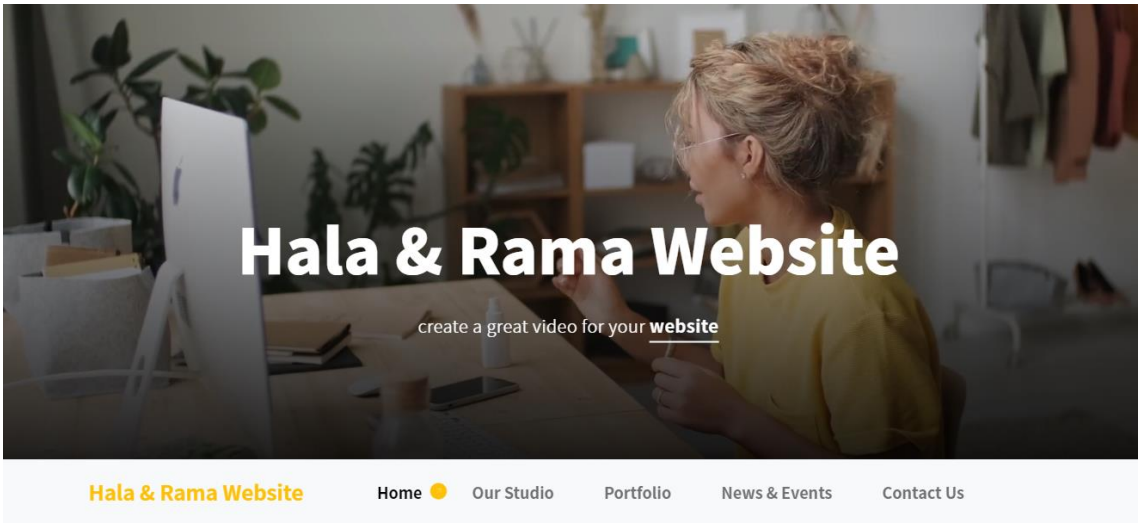
الشكل (11).

في حال قمنا بالاتصال من مكتبة الـ requests كذلك الامر يكون الخرج:

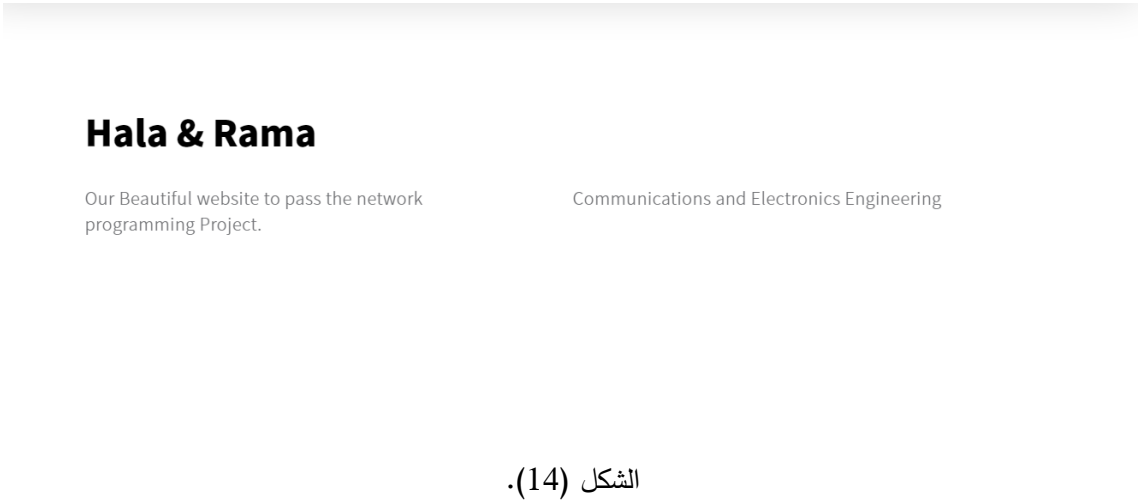
```
<html dir=rtl>
<head>
<meta charset="utf-8">
<title>
مشروع برمجة الشبكات
</title>
</head>
<body align=center bgcolor=#9EF760>
<h1>أهلا بكم في مشروع برمجة الشبكات</h1>
<h2>ا المشروع من إعداد الطلاب التالية أسمائهم
</h2>
<h3>2273 راما خالد حداد، 2059 حلا محمود علي

</h3>

</body>

</html>
>>
```

الشكل (12).

نقوم بتحميل قالب من موقع https://templatemo.com/ ونعدل عليه وهذا القالب يحوي ملفات CSS وملفات Java Scripts و Bootstrap وصور وفيديو ونقوم بالتعديل عليه بما يتناسب مع مشروعنا ونعرض لكم صور منه:



الشكل (13).



الشكل (14).

الشكل (15).

5. الاستنتاجات والتوصيات:

HTTPهو بروتوكول قابل للتوسيع سهل الاستخدام. هيكل خادم العميل، جنبًا إلى جنب مع القدرة على إضافة رؤوس، يسمح لـ HTTP بالتقدم جنبًا إلى جنب مع القدرات الموسعة للويب.

على الرغم من أن HTTP / 2 يضيف بعض التعقيد عن طريق تضمين رسائل HTTP في الإطارات لتحسين الأداء، إلا أن البنية الأساسية للرسائل ظلت كما هي منذ HTTP / 1.0. يظل تدفق الجلسة بسيطًا، مما يسمح بالتحقيق فيه وتصحيح أخطائه باستخدام جهاز مراقبة رسائل HTTP بسيط.

6.المراجع:

[1] https://www.techtarget.com/whatis/definition/HTTP-Hypertext-Transfer-Protocol

[2] Volker Turau "HTTPExplorer: exploring the hypertext transfer protocol" TiCSE '03: Proceedings of the 8th annual conference on Innovation and technology in computer science education

[3] http://www.steves-internet-guide.com/http-basics/
Understanding HTTP Basics -Updated:November 28, 2021

[4] https://www.arimetrics.com/en/digital-glossary/

[5] https://www.arimetrics.com/en/digital-glossary/