**📄 مستند SignalR — به زبان ساده**

**🔹 SignalR چیست؟**

SignalR یک کتابخانه توسعه داده شده توسط شرکت مایکروسافت است که به برنامه‌های تحت وب این امکان را می‌دهد تا با هم در زمان واقعی (real-time) صحبت کنند.

**💬 به عبارت ساده:**

وقتی کاربر A یک پیام بفرسته، کاربر B بدون نیاز به ریفرش کردن صفحه، فوراً پیام رو دریافت می‌کنه.

**🛠️ چطور کار می‌کند؟**

SignalR برای ارتباط real-time از چند روش مختلف استفاده می‌کند و به صورت خودکار بهترین روش موجود را انتخاب می‌کند:

| **روش** | **نوع ارتباط** | **توضیح** |
| --- | --- | --- |
| WebSocket | دوطرفه | سریع‌ترین روش، اجازه می‌دهد سرور و مرورگر همزمان با هم داده رد و بدل کنند. |
| Server-Sent Events (SSE) | یک طرفه (سرور به مرورگر) | وقتی فقط سرور باید چیزی بفرسته و مرورگر نیازی نداره داده بفرسته. |
| Long Polling | یک طرفه / تکراری | اگر روش‌های بالا پشتیبانی نشه، SignalR از این روش قدیمی ولی قابل اطمینان استفاده می‌کنه. |

**🧩 SignalR چگونه تصمیم می‌گیرد؟**

SignalR به صورت خودکار و هوشمندانه:

* وضعیت مرورگر (قدرت پشتیبانی از WebSocket و SSE)
* تنظیمات سرور
* شبکه و محیط کاربری

را بررسی می‌کند و بهترین روش موجود را انتخاب می‌کند.

**📌 کاربردهای SignalR**

* 💬 چت زنده
* 🚨 اعلان‌های لحظه‌ای (مثل "علی آنلاین شد")
* 💹 آپدیت قیمت‌ها (بورس، ارز)
* 🎮 بازی‌های آنلاین
* 📊 داشبوردهای زنده

**⚙️ مثال عملی: نحوه کار SignalR**

فرض کنید یک برنامه چت داریم:

1. کاربر A یک پیام می‌نویسه.
2. SignalR پیام رو به سرور می‌فرسته.
3. سرور به صورت خودکار پیام رو به کاربر B می‌ده.
4. کاربر B بدون نیاز به ریفرش، پیام رو می‌بینه.

**📈 مقایسه روش‌های ارتباطی**

| **ویژگی** | **WebSocket** | **Server-Sent Events** | **Long Polling** |
| --- | --- | --- | --- |
| دوطرفه | ✅ | ❌ | ❌ |
| یک طرفه | ✅ | ✅ | ✅ |
| سرعت | ⚡ بالا | ⚡ متوسط | 🐢 پایین |
| پشتیبانی در IE | ✅ (IE10+) | ❌ | ✅ |
| مناسب برای چت | ✅ | ❌ | ⚠️ در صورت نیاز فوری نه |
| مناسب برای آپدیت قیمت | ✅ | ✅ | ✅ |

**📝 نتیجه گیری**

SignalR یک ابزار فوق‌العاده برای ایجاد ارتباط زنده در برنامه‌های وب است.  
با استفاده از آن می‌توانید:

* بدون دغدغه روش ارتباطی، برنامه‌نویسی کنید.
* به کاربرانتون تجربه بهتری بدید.
* به‌روزرسانی‌ها را بدون نیاز به ریفرش صفحه انجام بدید.

قسمت دو  
  
**📚 مستند: Hub در SignalR**

**1️⃣ Hub چیه؟**

Hub در SignalR مثل "اتاق گفتگو" است که کلاینت (مرورگر) و سرور می‌تونن از طریق آن با هم داده رد و بدل کنن.

**💬 به عبارت ساده:**

اگر برنامه شما نیاز داره در زمان واقعی با سرور صحبت کنه (مثل فرستادن پیام چت یا گرفتن آپدیت لحظه‌ای)، از Hub استفاده می‌کنیم.

**2️⃣ چه متدهای مهمی داره؟**

دو متد خودکار وجود دارن که وقتی کاربر به hub وصل یا قطع می‌شه، صدا زده می‌شوند:

**🔹 OnConnected()**

* وقتی کاربر جدیدی به hub وصل می‌شه، این متد صدا زده می‌شه.
* مثلاً می‌تونیم اسم کاربر رو ثبت کنیم یا به دیگران بگیم "علی وارد شد".

**🔹 OnDisconnected(bool stopCalled)**

* وقتی کاربر از hub قطع می‌شه، این متد صدا زده می‌شه.
* مثلاً می‌تونیم به دیگران بگیم "علی خارج شد".

⚠️ نکته: این متدها اختیاری هستن و فقط برای مدیریت وضعیت کاربران استفاده می‌شن.

**3️⃣ تفاوت Clients, Groups, و Context**

| **عنصر** | **معنی** | **کاربرد** |
| --- | --- | --- |
| Clients | ابزاری برای فرستادن داده به کلاینت(ها) | مثلاً فرستادن پیام بههمهیا فقطیک کاربر |
| Groups | ابزاری برای فرستادن داده به گروهی از کلاینت‌ها | مثلاً فرستادن پیام به اعضای یک اتاق چت |
| Context | شامل اطلاعات ارتباطی فعلی | مثلاً گرفتن ID کاربر، IP، یا headerهای مرورگر |

**✅ مثال عملی:**

**فرض کنیم یک hub داریم به اسم ChatHub**



**👇 توضیح:**

* **Clients.All** = فرستادن پیام به همه کاربران.
* **Clients.User("Ali")** = فقط به کاربر Ali.
* **Clients.Group("Room1")** = فقط به اعضای گروه Room1.
* **Context.User.Identity.Name** = نام کاربری کسی که وصل شده.