

Conectando Lemmid Store con cocinas

Integrándose con sistemas externos que no controlas

Willem L. Middelkoop

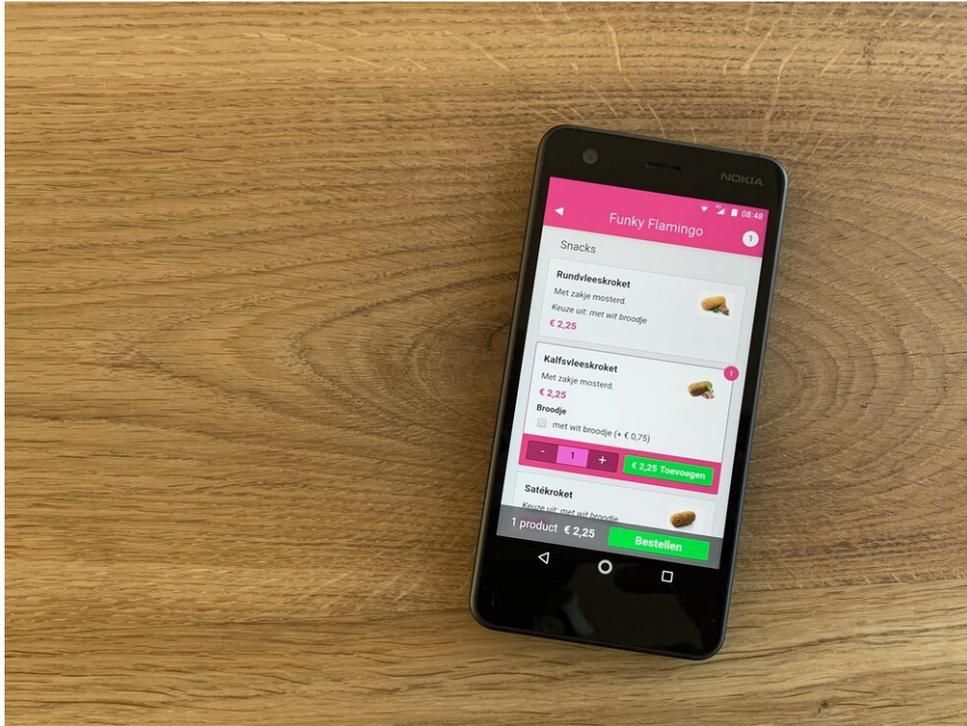
June 12, 2020

| Stks | Stks | Omschrijving | Prijs | Ttl |
|------|------|------------------|-------|-------|
| 1 | | Familiezak friet | 15.00 | 15.00 |
| | 1 | 6 personen | 0.00 | 0.00 |
| 4 | | Kipnuggets | 3.75 | 15.00 |
| 1 | | Satékroket | 2.25 | 2.25 |

Como parte de la aplicación de pedidos de comida que estoy desarrollando, necesitaba diseñar una forma confiable de vincular la aplicación a sistemas externos. Estos sistemas externos están fuera de mi control directo e incluyen diferentes cajas registradoras, sistemas de gestión de cocina e impresoras de tickets. Sigue leyendo para obtener más información sobre cómo diseñar para lo desconocido y poco confiable.

Aplicación para pedir comida

La aplicación que estoy desarrollando es [Lemmid Store](#), es una aplicación fácil de usar que permite a las personas realizar (y pagar) pedidos de comida para llevar. Puedes encontrar [all posts on Lemmid Store here](#).



Lemmid Store - una aplicación para pedir comida para llevar

Sistemas externos

La necesidad de conectar con sistemas externos está impulsada por el deseo de integrar la aplicación de pedidos de comida en los procesos comerciales relacionados con el servicio de entrega de alimentos. Piense en cosas como:

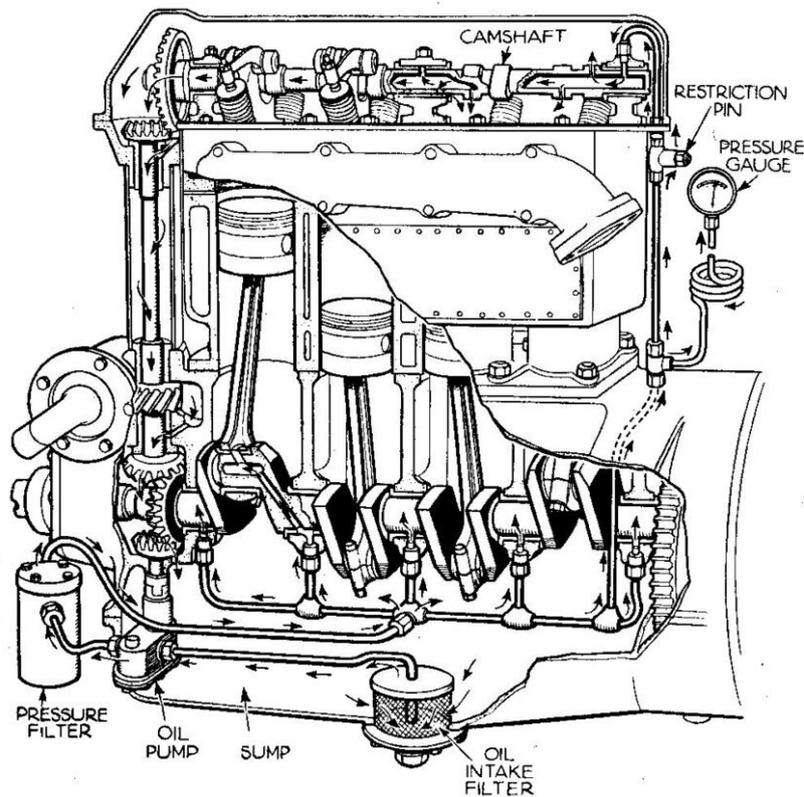
- recopilar datos de pedidos en registros administrativos (p. ej., contabilidad)
- enviar pedidos a las cocinas, donde se preparan y empaquetan las comidas
- proporcionar fácil acceso a la información de enrutamiento, mostrando dónde se debe entregar la comida
- programar pedidos de comida que están destinados a entregas posteriores

| Afhalen | | | | |
|----------------|------|------------------------------|-------|-------|
| [Redacted] | | | | |
| Ref: BEA5EA | | | | |
| Besteld: 16:35 | | Leveren: 2020-06-12 17:30:42 | | |
| Stks | Stks | Omschrijving | Prijs | Ttl |
| 1 | | Familiezak friet | 15.00 | 15.00 |
| | 1 | 6 personen | 0.00 | 0.00 |
| 4 | | Kipnuggets | 3.75 | 15.00 |
| 1 | | Satékroket | 2.25 | 2.25 |

Ticket impreso - a través de una conexión externa - incluyendo datos detallados del pedido como modificadores de producto (elecciones del cliente por producto)

Desafíos al conectar con sistemas externos

Puede que no te des cuenta, pero conectarse a sistemas externos puede ser complicado. Si piensas en la aplicación de pedidos como una especie de "motor" (como el de tu coche), esencialmente quieres integrar este motor con piezas externas. Como puedes imaginar, no todas las piezas son igualmente adecuadas para caber bajo el capó figurativo. Para hacer las cosas aún más desafiantes: no sabes con certeza cuán confiables son las partes externas. Debes diseñar tu integración de tal manera que diferentes partes externas se puedan conectar fácilmente y que el rendimiento sea independiente de la confiabilidad de estas partes externas.

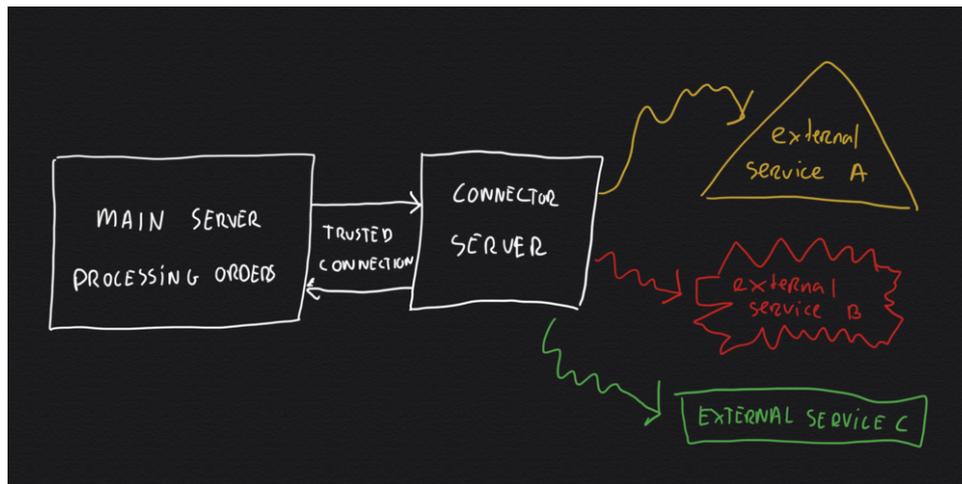


Integrar sistemas externos en tu aplicación es como adaptar piezas de terceros a un motor: debes considerar cuidadosamente su ajuste y confiabilidad (imagen de un motor de combustión, dominio público)

Al igual que con un motor de combustión, debes considerar la compatibilidad y la confiabilidad al conectar e integrar tu software con sistemas externos. **No puedes asumir que el software externo:**

- encaja perfectamente
- es siempre consistente
- está siempre disponible/presente
- es confiable

Para diseñar para esta visión bastante sombría de las conexiones externas, pensé que era una buena idea dejar que el servidor backend principal de mi aplicación de pedidos de comida descargara la responsabilidad de conectarse con sistemas externos a un 'servidor conector' especialmente diseñado. El servidor conector se conectaría entonces de forma independiente al sistema externo. De esta manera, el servidor principal solo se comunica con partes y sistemas de 'primera mano', minimizando los riesgos de incompatibilidades y posibles tiempos de inactividad.



Separando el servidor backend principal de los sistemas externos introduciendo un 'servidor conector' diseñado específicamente

En la práctica, esto significa que el 'servidor conector' es en realidad una máquina diferente, que ejecuta su propia pila de software. Hay algunas ventajas importantes en esta segmentación:

- **seguridad:** el servidor conector no mantiene un registro completo de todos los pedidos, solo aquellos pedidos que deben reenviarse a un sistema externo se mantienen en la memoria temporalmente. Un posible problema de seguridad en un sistema externo solo expondrá una cantidad mínima de datos.
- **se adapta a todo:** el servidor conector puede adaptarse a muchos protocolos y formas diferentes de conectarse a sistemas externos. Las conexiones a sistemas externos se pueden agregar de forma similar a como el servidor 'aprende un nuevo idioma para hablar'. Como ya es 'multilingüe', agregar un nuevo dialecto o idioma a menudo es fácil. Conectarse a diferentes API externas es más fácil cuando tienes todos los componentes básicos para (re)estructurar los datos.
- **se adapta a la falta de confiabilidad:** un simple mecanismo de cola en el servidor conector le permite manejar con elegancia posibles tiempos de inactividad y problemas con las conexiones externas. Al registrar y poner en cola por servicio externo, es posible minimizar el impacto en otras partes y conexiones cuando un servicio deja de funcionar.

Conclusión

Por muy tentador que sea simplemente agregar piezas nuevas y elegantes a tu aplicación, debes considerar cuidadosamente tu configuración para asegurarte de que funcione de manera confiable.

Diseñar software de servidor backend para conectarse con API externas es un desafío. ¡Debes esperar lo mejor, pero diseñar para lo peor!