

## Success Story

# Wetter.com

## Symfony2, le rayon de soleil de la météo allemande

Client : **Wetter.com**

Secteur d'activité : **Site de prévisions météorologiques**

Groupe : **Wetter.com AG**

Siège social : **Singen, Allemagne**

Taille : **20 salariés**

Site : **www.wetter.com**

Mots clés : **Symfony2, Twig, Varnish, CMS**



**Site ouvert au public** en 2001

**1er portail météo** en Allemagne

**656 millions** de visites en 2011

**13 220 000** de visiteurs uniques par mois

## Le projet en bref

Wetter.com, premier portail de prévisions météorologiques en Allemagne, a été développé il y a plus de dix ans. Face au vieillissement des technologies PHP utilisées à l'époque, il a été décidé, en 2011, de refondre complètement le site avec des outils et des langages plus avancés.

L'équipe de Wetter.com a ainsi choisi le Framework Symfony2 et plusieurs de ses composants pour la mise en œuvre d'un site plus souple, plus rapide, plus ergonomique et capable de supporter un trafic de plusieurs millions de visiteurs par mois. C'est la société 100 Days GmbH, basée à Stuttgart en Allemagne, qui a été chargée du développement et du lancement de cette application Symfony2 à grande échelle, avec le soutien des équipes de SensioLabs.

Les principaux bénéfices :

- ✓ Utilisation de Symfony2 comme outil d'agrégation d'informations issues de sources différentes (CMS, base de données de localisations, serveurs de cartes, etc.)
- ✓ Richesse et fiabilité des composants Symfony2 pour le développement d'un projet à grande échelle :
  - ✓ Performances du moteur de templates Twig
  - ✓ Complémentarité avec les outils de mise en cache (i.e. Varnish)
- ✓ Simplicité de prise en main de Symfony2, grâce notamment à une documentation complète qui facilite et accélère les développements d'applications

## Les enjeux

Fort de plusieurs millions de visiteurs uniques par mois, Wetter.com est le site leader de prévisions météorologiques en Allemagne. Développé il y a plus de dix ans, le site commençait cependant à montrer des signes de vieillissement, tant dans son interface obsolète que dans les technologies PHP élémentaires utilisées à l'époque. Il s'avérait donc nécessaire, pour les équipes de Wetter.com, de refondre le site pour en améliorer les services et en moderniser l'interface. Tout en assurant une migration souple et transparente.

L'enjeu était donc de taille pour 100 Days GmbH, société spécialisée dans le développement PHP basée à Stuttgart en Allemagne, en charge du développement en Symfony2, le framework retenu pour ce projet. « **Quand nous avons démarré le projet en 2011, Symfony2 était encore en version bêta,** » explique Andreas Streichardt, lead developer chez 100 Days et consultant sur le projet Wetter.com. « **Nous avons de l'expérience avec Symfony 1, mais c'était notre premier grand projet avec la nouvelle version** ».

Un grand projet qui devait en outre inclure un certain nombre de fonctionnalités très complexes : nouveau moteur de recherche par localisation, affichage des cartes et des données météo, gestion de bannières publicitaires, gestion de profils d'utilisateurs, etc. Le tout sur un site entièrement éditable via un CMS.



“ Les nombreux composants que nous avons utilisés pendant ce projet sont simples et bien documentés. Symfony2 permet de développer des applications très facilement, y compris pour des projets à grande échelle. ”

Andreas Streichardt, Consultant Symfony2 pour Wetter.com

## La solution

Pour relever ce défi, 100 Days s'est naturellement tourné, dans un premier temps, vers SensioLabs pour la prise en main de Symfony2. « **Les équipes françaises de SensioLabs nous ont beaucoup aidés,** » se souvient Andreas Streichardt. « **Experts dans leur domaine, tous les consultants que nous avons rencontrés ont été très présents à nos côtés, ont toujours répondu rapidement à nos questions : nous avons ainsi gagné beaucoup de temps dans le projet.** »



Pour Wetter.com, l'objectif était de mettre en œuvre une architecture capable de supporter des flux d'informations en provenance de plusieurs sources. Symfony2 a ainsi permis d'agréger, au sein des pages du site, tous les contenus et layouts issus du CMS, toutes les données de localisations (villes, pays...) issues de la base de données, et surtout toutes les données issues du serveur de cartes et de prévisions météorologiques. « **Symfony2 simplifie fondamentalement la connexion aux données externes, où qu'elles se trouvent** » explique Gaylord Aulke, fondateur de 100 Days.

Dans ce projet, 100 Days a également utilisé de nombreux composants de Symfony, notamment Twig, « **le meilleur moteur de templates** », selon Andreas Streichardt. « **Twig est bien codé, rapide et facilement extensible, tandis que le système de "blocs" permet vraiment de mieux organiser les pages** » poursuit-il. Tandis que la gestion des caches, native et sans code spécifique dans Symfony2, et optimisée grâce à l'utilisation de Varnish (prise en charge du langage ESI), était indispensable pour un site de cette ampleur.

## Les bénéfices

La nouvelle version du site Wetter.com a été mise en ligne en mars 2012, et très vite plébiscitée par les internautes. Son interface a été considérablement modernisée et ses fonctionnalités donnent au site une nouvelle dynamique. Déploiement de cartes de différentes zones géographiques, personnalisation de l'interface via un système de favoris, génération automatique de contenus, gestion automatique des widgets... : Symfony2 a permis, en découplant les différents composants, de mettre en œuvre un site particulièrement riche et simple d'utilisation, tant côté back-end que front-end.

Pour 100 Days, l'expérience a également été positive : « **Nous avons pris beaucoup de plaisir à utiliser Symfony2 et nous avons énormément appris pendant ce projet** », se réjouit Andreas Streichardt. « **Les nombreux composants que nous avons utilisé sont simples et bien documentés. Symfony2 permet de développer des applications très facilement, y compris pour des projets à grande échelle** », conclut-il.



+33 (0) 1 40 99 81 09



sales@sensiolabs.com

# SensioLabs

Créateur de  Symfony