BLACKBIRD FOR TRAINING

دورة بايثون وDjangoo للمستوى المتوسط: تطوير الواجهات الخلفية وواجهات REST وتحسين الأداء

8 – 12 فبراير 2027

أمستردام (هولندا)

UK Traininig

PARTNER

www.blackbird-training.com

دورة بايثون وDjangog للمستوى المتوسط: تطوير الواجهات الخلفية وواجهات REST وتحسين الأداء

المرجع: 31_361605 **التاريخ:** 8 – 12 فبراير 2027 **المكان:** أمستردام (هولندا) **الرسوم:** GBP 5900

مقدمة:

دورة بايثون وDjangog للمستوى المتوسط تتناول تطوير الواجهات الخلفية باستخدام Django، إطار عمل الويب بلغة بايثون، Djangog REST Framework، مجموعة الأدوات المخصصة لبناء واجهات REST، وهي واجهات برمجة تطبيقات تتبادل البيانات بصيغة JSON عبر بروتوكول HTTP. تمتد هذه الدورة من بلاك بيرد للتدريب أسبوع، وتتوجه إلى مطوري البرمجيات ومطوري الواجهات الخلفية والويب ومبرمجي بايثون الذين يبنون تطبيقات الويب أو يصونونها، ويخرج منها المشاركون بواجهة Django REST مختبرة وموثقة مع تقرير لضبط الأداء. وتقدم بمستوى تطبيقي متوسط عبر مختبرات عملية مبنية على سيناريوهات مشاريع واقعية، من نمذجة البيانات والمهام الخلفية باستخدام Celery إلى الاختبار باستخدام pytest واختبار الحمل.

أهداف الدورة:

- تطبيق الدوال والأصناف والبيئات الافتراضية في بايثون وقواعد HTTP لبناء نقاط نهاية في Django REST Framework تعيد رموز الحالة الصحيحة وبيانات JSON
- التمييز بين المفاتيح الأساسية والمفاتيح الأجنبية، واختيار أنواع العلاقات وعمليات الربط في SQL والمسلسلات المتداخلة الملائمة لنموذج بيانات علائقي
- تطبيق منطق الأعمال وطوابير الرسائل في Redis والمهام الخلفية في Celery، والتمييز بين إشارات Django والمزخرفات عند معالجة أحداث التطبيق
- المقارنة بين الاختبارات الأحادية واختبارات التكامل، وتطبيق pytest وأدوات الاختبار في Djangog factory_boy لبناء حزمة اختبارات قابلة للتكرار
- تفسير توثيق Swagger/OpenAPI ومخرجات استعلامات ORM لاختيار تحسينات التخزين المؤقت وقاعدة البيانات لنقاط النهاية البطيئة
- تبرير قرارات ضبط الأداء بنتائج اختبار الحمل المسجلة في تقرير ضبط أداء الواجهة

الفئة المستهدفة:

- مطورو الواجهات الخلفية الذين يبنون خدمات Django ويوسعونها، ويقررون أين يوضع منطق الأعمال والمهام الخلفية
- مطورو الويب الذين يتيحون البيانات لتطبيقات الواجهة الأمامية والهواتف، ويختارون طريقة تصميم نقاط النهاية والمسلسلات ورموز الحالة
- مبرمجو بايثون المنتقلون إلى تطوير الويب، الذين يقدرون أي أنماط Django REST Frameworkg Django تناسب مشروعا جديدا
- مهندسو البرمجيات الذين يصونون تطبيقات ويب قائمة، ويقررون أي الاستعلامات البطيئة والذاكرة المؤقتة ونقاط النهاية يعالجونها أولا
- مختصو تقنية المعلومات الذين يدعمون تطبيقات الويب وواجهات برمجة التطبيقات ويختبرونها، ويقدرون متى تكون تغطية الاختبارات والتوثيق جاهزة للإطلاق

محاور الدورة:

اليوم 1: أساسيات Django وتطوير واجهات REST

- الدوال مقابل الأصناف في بايثون: هيكلة شيفرة خلفية قابلة لإعادة الاستخدام في مشاريع Django
- البيئات الافتراضية وإدارة الحزم: pipg venv وملفات المتطلبات
- طرق HTTP ورموز الحالة: GET وPOST وPUT وPATCH وDELETE في واجهات برمجة التطبيقات
- دورة حياة الطلب والاستجابة وصيغة JSON: كيف تعالج الوسائط والعروض في Django الاستدعاءات
- أساسيات Django REST Framework: APIView وViewSetsg والموجهات وأول نقاط نهاية للواجهة

اليوم 2: نمذجة قواعد البيانات والواجهات المتداخلة

- تصميم قواعد البيانات العلائقية: الجداول والتطبيع والعلاقات بين الكيانات في تطبيقات الويب
- المفاتيح الأساسية مقابل المفاتيح الأجنبية: نمذجة علاقات واحد إلى متعدد ومتعدد إلى متعدد
- أساسيات SQL: استعلامات SELECT وINNER JOIN وLEFT JOIN على الجداول المرتبطة
- نماذج Djangoerg ModelSerializer: ربط جداول قاعدة البيانات ببيانات واجهة متحقق منها
- المسلسلات المتداخلة: البيانات المتداخلة القابلة للكتابة وعمليات الإنشاء والتحديث المرتبطة

اليوم 3: منطق الأعمال والمعالجة غير المتزامنة

- منطق الأعمال في Django: طبقات الخدمات ودوال النماذج وقواعد التحقق
- أساسيات Redis وطوابير الرسائل: الوسطاء والمفاتيح وهياكل البيانات المستخدمة للمهام
- معالجة المهام الخلفية باستخدام Celery: العمال وإعادة المحاولة والمهام الدورية عبر Beat
- إشارات Django مقابل المزخرفات: معالجة الأحداث بشكل منفصل وإعادة استخدام سلوك العروض
- البرمجة غير المتزامنة وأنظمة الإشعارات: العروض غير المتزامنة وتنبيهات البريد والإشعارات الفورية

اليوم 4: الاختبار وتوثيق واجهات برمجة التطبيقات

- الاختبارات الأحادية مقابل اختبارات التكامل: النطاق والعزل والتغطية في واجهات Django
- pytest pytest-djangog: التجهيزات والاختبارات متعددة المعاملات والوصول إلى قاعدة البيانات أثناء الاختبار
- أدوات الاختبار في Django: TestCase APIClientg APITestCaseg لفحص نقاط النهاية
- إدارة بيانات الاختبار باستخدام factory_boy: المصانع SubFactoryg وبيانات تجهيز واقعية
- توثيق الواجهات باستخدام Swagger/OpenAPI: مخططات drf-spectacular والاختبار التفاعلي لنقاط النهاية

اليوم 5: المختبر: تحسين الأداء وقابلية التوسع

- select_related مقابل prefetch_related: الاستخدام المتقدم لأداة Django ORM لمعالجة مشكلة استعلامات N+1
- تحسين الاستعلامات باستخدام Django Debug Toolbar: تحليل الاستعلامات والتجميعات وفهارس قاعدة البيانات
- استراتيجيات التخزين المؤقت: على مستوى العرض وأجزاء القوالب والمستوى المنخفض باستخدام Redis
- اعتبارات أداء قاعدة البيانات: تجميع الاتصالات وتقسيم النتائج إلى صفحات والعمليات المجمعة
- اختبار الحمل باستخدام Locust وتحسين أداء الواجهات: استكمال تقرير ضبط الأداء

المهارات التي ستكتسبها:

- تصميم واجهات REST
- نمذجة البيانات العلائقية
- بناء المسلسلات المتداخلة
- معالجة المهام غير المتزامنة
- الاختبار الآلي للواجهات
- توثيق OpenAPI
- تحليل استعلامات ORM
- تحليل نتائج اختبار الحمل

لماذا تحضر هذه الدورة:

- من كتابة عروض Django بشكل متفرق إلى هيكلة نقاط نهاية Django REST Framework برموز حالة ومسلسلات وموجهات متسقة
- من تنفيذ الأعمال البطيئة داخل طلبات الويب إلى نقلها إلى عمال Celery مع Redis وسيطا للرسائل
- من الفحص اليدوي لنقاط النهاية إلى حزمة اختبارات آلية في pytest ببيانات factory_boy وتوثيق Swagger/OpenAPI
- من التخمين في أسباب بطء الصفحات إلى واجهة Django REST مختبرة وموثقة مع تقرير ضبط أداء مدعوم ببيانات اختبار الحمل

الخاتمة:

تطوير الواجهات الخلفية ببايثون Djangog يعني تحويل نماذج البيانات إلى خدمات ويب موثوقة ومختبرة تحافظ على سرعتها مع نمو حركة الاستخدام. وتوضح الدورة ثلاثة فروق أساسية: المفاتيح الأساسية مقابل المفاتيح الأجنبية في النمذجة العلائقية، وإشارات Django مقابل المزخرفات عند الاستجابة للأحداث، والاختبارات الأحادية مقابل اختبارات التكامل في مشروع REST Framework Django. وهي مناسبة للمطورين ومبرمجي بايثون الذين يكتبون شيفرة بايثون الأساسية ويحتاجون الآن إلى مهارة تطبيقية في بناء واجهات Django واختبارها وتحسين أدائها، لا إلى دورة تمهيدية في الويب ولا إلى دورة في معمارية الأنظمة الموزعة.

الأسئلة الشائعة:

ما الذي يحتاج المشارك إلى معرفته قبل دورة بايثون Djangog للمستوى المتوسط؟

يحتاج المشارك إلى معرفة عملية بأساسيات لغة بايثون وخبرة أولية في كتابة شيفرة لتطبيقات الويب. وتراجع الدوال والأصناف والبيئات الافتراضية في اليوم الأول، فلا يشترط استخدام سابق ل Django REST Framework أو Celery أو pytest.

بماذا تختلف دورة بايثون Djangog للمستوى المتوسط عن دورة تمهيدية في تطوير الويب؟

تركز الدورة التمهيدية على القوالب والنماذج والصفحات الأولى، أما هذه الدورة فتركز على الواجهات الخلفية: واجهات REST باستخدام Django REST Framework، والنمذجة العلائقية، والمهام الخلفية عبر Celery Redisg، والاختبار الآلي، وتحسين أداء التطبيقات القائمة.

ما الفرق بين select_related prefetch_relatedg في تحسين أداء بايثون Djangog؟

تتبع select_related علاقات المفتاح الأجنبي وواحد إلى واحد عبر عملية JOIN في استعلام واحد، بينما تنفذ prefetch_related استعلاما منفصلا لعلاقات متعدد إلى متعدد والعلاقات العكسية ثم تربط النتائج داخل بايثون. وكلتاهما تعالج مشكلة استعلامات N+1 [illegible] المناسب.

UK Training
PARTNER

ماذا يعود المشارك به إلى عمله من دورة بايثون وDjangog للمستوى المتوسط؟

يعود المشارك بمشروع واجهة Django REST مختبرة وموثقة، يضم حزمة اختبارات في pytest factory_boyg وتوثيق Swagger/OpenAPI وتقريرا لضبط الأداء مبنيا على نتائج اختبار الحمل، جاهزا ليكون مرجعا لتطبيقاته الخاصة.